

**UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS**

ALEXANDRE PERGHER AMARO

**DESENVOLVIMENTO DE UM CHATBOT INTEGRADO AO SISTEMA APS
DE UMA INDÚSTRIA DO SETOR METALOMECÂNICO**

CAXIAS DO SUL

2020

ALEXANDRE PERGHER AMARO

**DESENVOLVIMENTO DE UM CHATBOT INTEGRADO AO SISTEMA APS
DE UMA INDÚSTRIA DO SETOR METALOMECÂNICO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientadora: Profa. Ma. Iraci Cristina da Silveira De Carli

CAXIAS DO SUL

2020

ALEXANDRE PERGHER AMARO

**DESENVOLVIMENTO DE UM CHATBOT INTEGRADO AO SISTEMA APS
DE UMA INDÚSTRIA DO SETOR METALOMECÂNICO**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em 02/12/2020

Banca Examinadora

Profa. Ma. Iraci Cristina da Silveira De Carli
Universidade de Caxias do Sul - UCS

Profa. Dra. Carine Geltrudes Webber
Universidade de Caxias do Sul - UCS

Prof. Esp. Daniel Antônio Faccin
Universidade de Caxias do Sul - UCS

RESUMO

O objetivo geral deste trabalho de conclusão de curso consiste em desenvolver um *chatbot* que seja capaz de responder as perguntas mais frequentes sobre os pedidos de venda de uma determinada indústria do setor metalomecânico. O *chatbot* desenvolvido também foi integrado ao sistema *Drummer APS*, que já era utilizado pela organização em questão, por meio das técnicas de acesso direto ao banco de dados *Oracle* e execução de transações específicas por linha de comando. Para atingir esse objetivo foram utilizados alguns serviços de inteligência artificial do kit de desenvolvimento de *software Microsoft Bot Framework*, tal como, o *LUIS* para que o *chatbot* pudesse interagir naturalmente com o usuário e o Serviço de *Bot* do *Azure* para desenvolver, testar, implantar e publicar o aplicativo *web* no canal *Microsoft Teams*. Ao final do trabalho, foram realizados dois pilotos *in loco* e uma pesquisa qualitativa *on-line* com os usuários chave para validar se os objetivos do trabalho foram atingidos, identificar os principais benefícios da solução apresentada, coletar sugestões de melhoria e vislumbrar oportunidades para trabalhos futuros.

Palavras-chave: Inteligência Artificial. Processamento de Linguagem Natural. *Chatbot*. Sistema de Planejamento Avançado de Produção. Integração de Sistemas.

ABSTRACT

The general objective of this course completion work is to develop a chatbot that is able to answer the most frequently asked questions about sales orders from a particular industry of the metalworking sector. The developed chatbot was also integrated with the Drummer APS system, which was already used by the organization, through the techniques of direct access to the Oracle database and execution of specific transactions by command line. To achieve this goal, some artificial intelligence services of the Microsoft Bot Framework software development kit were used, such as LUIS so that the chatbot could interact naturally with the user and the Azure Bot Service to develop, test, deploy, and publish the web application to the Microsoft Teams channel. At the end of the work, two pilots were conducted on site and an online qualitative research with key users to validate whether the work objectives were achieved, identify the main benefits of the solution presented, collect suggestions for improvement and envision opportunities for future work.

Keywords: *Artificial Intelligence. Natural Language Processing. Chatbot. Advanced Planning and Scheduling System. Systems Integration.*

LISTA DE FIGURAS

Figura 1 – Funções de um SI	44
Figura 2 – Dimensões de um SI	45
Figura 3 – Abrangência dos diferentes tipos de MRP	47
Figura 4 – Estrutura conceitual do ERP e sua evolução desde o MRP	48
Figura 5 – Estrutura típica de um sistema ERP	48
Figura 6 – Fluxo de PPCPM.....	50
Figura 7 – Fluxo básico de integração de sistemas	51
Figura 8 – Elementos da computação orientada a serviços	53
Figura 9 – Exemplo de comunicação via RPC	55
Figura 10 – Exemplo de comunicação via MOM	55
Figura 11 – Diagrama de atividades sem <i>chatbot</i>	61
Figura 12 – Diagrama de atividades com <i>chatbot</i>	62
Figura 13 – Arquitetura do <i>Microsoft Bot Framework</i>	65
Figura 14 – Exemplo de funcionamento do LUIS em um <i>chatbot</i>	66
Figura 15 – Exemplo de funcionamento do <i>QnA Maker</i> em um <i>chatbot</i>	66
Figura 16 – Exemplo de funcionamento do LUIS e do <i>QnA Maker</i> combinados.....	67
Figura 17 – Diagrama de implantação do <i>chatbot</i>	69
Figura 18 – Aplicativo de conversa criado no LUIS	73
Figura 19 – Intenções cadastradas no LUIS	74
Figura 20 – Entidades cadastradas no LUIS	77
Figura 21 – Treinamento do modelo de linguagem no LUIS	78
Figura 22 – Teste do modelo de linguagem no LUIS	79
Figura 23 – Exemplo de teste do modelo de linguagem no LUIS.....	79
Figura 24 – Publicação do modelo de linguagem no LUIS.....	80
Figura 25 – Grupo de recursos criado no <i>Azure</i>	81
Figura 26 – Modelo do <i>bot</i> selecionado no <i>Azure</i>	81
Figura 27 – Exportação do modelo de linguagem no LUIS	82
Figura 28 – Comando para geração de classe C# do modelo de linguagem	82
Figura 29 – Tela de login do <i>Drummer APS</i>	83
Figura 30 – Visão de pedidos de venda criada no <i>Drummer APS</i>	84
Figura 31 – Exemplo de consulta de pedidos de venda no <i>Drummer APS</i>	85
Figura 32 – Classe <i>DBConnection</i>	85

Figura 33 – Processamento programado do <i>Drummer</i> APS.....	86
Figura 34 – Arquivo <i>batch</i> para processamento programado do <i>Drummer</i> APS	87
Figura 35 – Propriedades do arquivo <i>batch</i> do <i>Drummer</i> APS	87
Figura 36 – Estrutura do projeto do <i>chatbot</i> no <i>Microsoft Visual Studio</i>	88
Figura 37 – Método <i>IntroStepAsync</i> da classe <i>MainDialog</i>	89
Figura 38 – Classe Pedido.....	89
Figura 39 – Método <i>ActStepAsync</i> da classe <i>MainDialog</i>	90
Figura 40 – Método <i>FinalStepAsync</i> da classe <i>MainDialog</i>	91
Figura 41 – <i>Bot Framework Emulator</i>	91
Figura 42 – Tela de conexão do <i>Bot Framework Emulator</i>	92
Figura 43 – Exemplo de teste no <i>Bot Framework Emulator</i>	92
Figura 44 – Implantação do <i>chatbot</i> pela CLI do <i>Azure</i>	93
Figura 45 – Publicação do <i>chatbot</i> no canal <i>Microsoft Teams</i> pelo <i>Azure</i>	94
Figura 46 – Clonagem de ambiente no <i>Drummer</i> APS.....	95
Figura 47 – Exemplo de pergunta realizada pelo UC do PCP no piloto.....	95
Figura 48 – Exemplo de pergunta realizada pelo UC do Comercial no piloto	96
Figura 49 – Seção 1 do formulário de avaliação do <i>chatbot</i>	97
Figura 50 – Aprendizado supervisionado no LUIS.....	104

LISTA DE QUADROS

Quadro 1 – Atributos para classificação de <i>chatbots</i>	41
Quadro 2 – Matriz SIPOC com as perguntas mais frequentes sobre pedidos	59
Quadro 3 – Requisitos funcionais do <i>chatbot</i>	59
Quadro 4 – Requisitos não funcionais do <i>chatbot</i>	59
Quadro 5 – Matriz SIPOC agrupada pela saída esperada	70
Quadro 6 – Intenções e enunciados extraídos da matriz SIPOC	70
Quadro 7 – Intenções e enunciados do <i>chatbot</i>	74
Quadro 8 – Resultado dos pilotos do <i>chatbot</i>	96
Quadro 9 – Respostas da pesquisa qualitativa	111

LISTA DE ABREVIATURAS

abr.	abril
ago.	agosto
ampl.	ampliada
dez.	dezembro
Dr.	Doutor
Dra.	Doutora
ed.	edição
Esp.	Especialista
<i>et al.</i>	e outros
<i>etc.</i>	entre outros
f.	folhas
fev.	fevereiro
jan.	janeiro
jul.	julho
jun.	junho
Ltda.	Limitada
Ma.	Mestra
mar.	março
n.	número
nov.	novembro
org.	organizador
out.	outubro
p.	página
Prof.	Professor
Profa.	Professora
rev.	revisada
set.	setembro
Sr.	Senhor
Sra.	Senhora
v.	volume

LISTA DE SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
AI	<i>Artificial Intelligence</i>
AIML	<i>Artificial Intelligence Markup Language</i>
ALICE	<i>Artificial Linguistic Internet Computer Entity</i>
API	<i>Application Programming Interface</i>
APS	<i>Advanced Planning and Scheduling</i>
ASP	<i>Active Server Pages</i>
BD	Banco de Dados
C#	<i>C Sharp</i>
CLI	<i>Command-Line Interface</i>
CRP	<i>Capacity Requirements Planning</i>
DRP	<i>Distribution Requirements Planning</i>
EDI	<i>Electronic Data Interchange</i>
ERP	<i>Enterprise Resource Planning</i>
FAQ	<i>Frequently Asked Questions</i>
HTTP	<i>HyperText Transfer Protocol</i>
IA	Inteligência Artificial
IDE	<i>Integrated Development Environment</i>
IEC	<i>International Electrotechnical Commission</i>
ISO	<i>International Organization for Standardization</i>
JSON	<i>JavaScript Object Notation</i>
LUIS	<i>Language Understanding Intelligent Service</i>
MIT	<i>Massachusetts Institute of Technology</i>
ML	<i>Machine Learning</i>
MOM	<i>Message-Oriented Middleware</i>
MPS	<i>Master Production Schedule</i>
MRP	<i>Material Requirements Planning</i>
MRP II	<i>Manufacturing Resource Planning</i>
NBR	Norma Brasileira
NLP	<i>Natural Language Processing</i>
OC	Ordem de Compra

OF	Ordem de Fabricação
OM	Ordem de Montagem
OP	Ordem de Produção
PCP	Planejamento e Controle da Produção
PLN	Processamento de Linguagem Natural
PPCP	Planejamento, Programação e Controle da Produção
PPCPM	Planejamento, Programação e Controle da Produção e dos Materiais
PUR	<i>Purchasing</i>
RC	<i>Release Candidate</i>
RCCP	<i>Rough-Cut Capacity Planning</i>
REST	<i>Representational State Transfer</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
RPC	<i>Remote Procedure Call</i>
S&OP	<i>Sales and Operations Planning</i>
SDK	<i>Software Development Kit</i>
SFC	<i>Shop Floor Control</i>
SGBD	Sistema de Gerenciamento de Banco de Dados
SI	Sistema de Informação
SIPOC	<i>Supplier, Input, Process, Output, Customer</i>
SOA	<i>Service-Oriented Architecture</i>
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso
TI	Tecnologia da Informação
UC	Usuário Chave
UCS	Universidade de Caxias do Sul
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	21
1.1	PROBLEMA DE PESQUISA	22
1.2	QUESTÃO DE PESQUISA.....	24
1.3	OBJETIVOS	25
1.3.1	Objetivo geral.....	25
1.3.2	Objetivos específicos.....	25
1.4	METODOLOGIA.....	26
1.4.1	Fase I	26
1.4.2	Fase II.....	27
1.5	ESTRUTURA DO TEXTO	28
2	INTELIGÊNCIA ARTIFICIAL	31
2.1	TESTE DE TURING	31
2.2	ÁREAS DE APLICAÇÃO	32
2.2.1	Jogos.....	32
2.2.2	Raciocínio automático	32
2.2.3	Sistemas especialistas	33
2.2.4	Compreensão da linguagem natural.....	34
2.2.5	Modelagem do desempenho humano.....	34
2.2.6	Planejamento e robótica	34
2.2.7	Linguagens e ambientes para IA.....	35
2.2.8	Aprendizagem de máquina.....	35
2.2.9	Redes neurais e algoritmos genéticos	35
2.2.10	Filosofia.....	36
2.3	CHATBOTS.....	36
2.3.1	História dos chatbots	37
2.3.2	Atributos dos chatbots	39
2.3.3	Processamento de linguagem natural	41
2.4	CONSIDERAÇÕES FINAIS.....	42
3	INTEGRAÇÃO DE SISTEMAS	43
3.1	SISTEMAS DE INFORMAÇÃO	43
3.1.1	Sistema MRP	45
3.1.2	Sistema MRP II.....	46

3.1.3	Sistema ERP	47
3.1.4	Sistema APS	49
3.2	TÉCNICAS DE INTEGRAÇÃO	51
3.2.1	SQL	52
3.2.2	SOA	52
3.2.2.1	API	53
3.2.2.2	REST	53
3.2.2.3	API RESTful.....	54
3.2.3	RPC	54
3.2.4	MOM	55
3.2.5	EDI	56
3.3	CONSIDERAÇÕES FINAIS	56
4	ESPECIFICAÇÃO	57
4.1	DEFINIÇÃO DE USUÁRIOS CHAVE	57
4.2	ELICITAÇÃO DE REQUISITOS.....	58
4.3	PROJETO DE SOFTWARE.....	60
4.3.1	Modelagem comportamental	60
4.3.2	Componentes de software de terceiros	63
4.3.2.1	Drummer APS.....	63
4.3.2.2	Microsoft Bot Framework	64
4.3.2.2.1	<i>Serviços de PLN</i>	65
4.3.2.2.2	<i>Canais</i>	68
4.3.3	Modelagem estrutural	68
4.3.4	Modelagem linguística	69
5	IMPLEMENTAÇÃO	73
5.1	DESENVOLVIMENTO	73
5.1.1	Criação de aplicativo no LUIS	73
5.1.1.1	Cadastro das intenções no LUIS	74
5.1.1.2	Cadastro das entidades no LUIS	77
5.1.1.3	Treinamento do modelo de linguagem.....	78
5.1.1.4	Testes do modelo de linguagem	78
5.1.1.5	Publicação do modelo de linguagem	80
5.1.2	Criação de aplicativo no Azure	80
5.1.2.1	Integração do chatbot com o LUIS.....	82

5.1.2.2	Integração do chatbot com o Drummer APS	83
5.1.2.2.1	<i>Acesso direto à base de dados</i>	<i>83</i>
5.1.2.2.2	<i>Execução por linha de comando</i>	<i>85</i>
5.1.2.3	Codificação do chatbot.....	88
5.2	TESTES	91
5.3	IMPLANTAÇÃO.....	93
5.4	PUBLICAÇÃO	93
5.5	PILOTO	94
5.6	PESQUISA QUALITATIVA.....	97
6	CONCLUSÃO	101
6.1	AVALIAÇÃO DOS RESULTADOS	102
6.2	BENEFÍCIOS.....	102
6.3	TRABALHOS FUTUROS.....	103
	REFERÊNCIAS.....	107
	APÊNDICE A – RESPOSTAS DA PESQUISA QUALITATIVA.....	111

1 INTRODUÇÃO

A sociedade vive a era da inovação. Nesse novo mundo, a inteligência artificial (IA) assume um papel extremamente relevante no cotidiano das pessoas.

Graças ao advento eminente da tecnologia da informação (TI) na vida das pessoas, a IA já está marcando presença nas mais diversas áreas da sociedade, como por exemplo: na indústria, nos negócios, na saúde, na segurança, na educação, na agricultura, na comunicação, no marketing e na prestação de serviços.

Kai-Fu Lee (2019), um dos maiores especialistas em inovação tecnológica do mundo, afirma categoricamente que a IA mudará a forma como os seres humanos trabalham e se relacionam uns com os outros.

De fato, a IA tornou-se imprescindível para o sucesso de qualquer negócio, e é pensando nisso, que nos últimos anos, as empresas têm investido cada vez mais em máquinas e aplicações que implementam algoritmos de IA.

Uma das soluções de IA que está ganhando grande destaque no mercado global são os *chatbots*. A maioria desses robôs inteligentes de conversação utilizam algoritmos de Processamento de Linguagem Natural (PLN) para se comunicar com os usuários de maneira natural e, normalmente, automatizam rotinas de cadastro e consulta de informações.

Muitas vezes, os *chatbots* adotados como estratégia para impulsionar os negócios das empresas também são integrados aos mais variados sistemas de informação da organização, tornando-se ainda mais eficazes nas suas tarefas, e de modo a melhorar significativamente a experiência dos usuários.

Ainda, segundo pesquisa realizada pelo grupo *Gartner*, até o final de 2022, 70% dos executivos interagirão com plataformas de conversação diariamente, e os *chatbots* já são considerados pelos gestores de TI das grandes empresas como a principal aplicação baseada em IA da atualidade.¹

Pois bem, é diante desse cenário emergente e desafiador que o presente trabalho sugere a adoção de um *chatbot* integrado ao sistema APS (*Advanced Planning and Scheduling*) de uma indústria do setor metalomecânico para solucionar o problema de pesquisa elucidado na próxima seção.

¹ Informação extraída de: <https://www.gartner.com/smarterwithgartner/chatbots-will-appeal-to-modern-workers>.

1.1 PROBLEMA DE PESQUISA

O Grupo Sazi, fundado em 1984, é um tradicional conglomerado industrial do setor metalomecânico que possui quatro unidades de negócio: Sazi Máquinas, Sazi Laser, Maltec e Vega Maltec. Sua missão é transformar ideias e tecnologia em soluções inovadoras para o mercado, agregando valor aos clientes, aos profissionais internos, aos fornecedores, a sociedade e aos acionistas.²

Além disso, visando ser a melhor opção de fornecimento do seu segmento, é reconhecida como uma empresa de classe mundial. O grupo empresarial também já foi destaque em importantes premiações nacionais e internacionais.

Com sede corporativa na cidade de Farroupilha, no estado do Rio Grande do Sul, a empresa matriz, Máquinas Sazi Ltda., abriga a principal fábrica do grupo, que é responsável pelas unidades Sazi Máquinas, referência no mercado global de máquinas para calçados, e Sazi Laser, especialista na fabricação de peças com corte laser para os mais diversos segmentos do mercado.

Contudo, apesar de ser uma organização que investe constantemente em tecnologia de ponta e na melhoria contínua de seus processos, atualmente, os colaboradores do Grupo Sazi possuem diversas restrições e dificuldades para responder as perguntas mais frequentes dos clientes sobre pedidos de venda. Dentre os impedimentos encontrados para tal, pode-se citar:

- a) comunicação interna ineficaz via correspondência eletrônica;
- b) diversas mensagens eletrônicas perdidas e sem retorno;
- c) interdependência entre a área comercial e o setor de Planejamento e Controle da Produção (PCP);
- d) baixa disponibilidade dos profissionais para atendimento aos colegas;
- e) centralização do conhecimento organizacional;
- f) descentralização da informação em múltiplos sistemas, relatórios, planilhas e painéis de controle;
- g) algumas ferramentas com desempenho aquém do desejado;
- h) processos demasiadamente extensos e morosos;
- i) ausência de documentação para processos específicos, como por exemplo, o procedimento de aprazamento de pedidos;

² Vide mais informações sobre o Grupo Sazi em: <http://www.sazi.com.br>.

- j) poucos funcionários com total domínio das habilidades exigidas para análise de dados nos diversos sistemas de gestão utilizados pela empresa;
- k) elevado nível de compreensão do negócio indispensável para responder questões mais complexas;
- l) dependência do setor de Tecnologia da Informação (TI) para embasamento de respostas e realização de consultas avançadas via SQL (*Structured Query Language*).

Um exemplo para compreender melhor esse cenário é que, diariamente, os técnicos comerciais respondem aos questionamentos dos clientes referentes a possibilidade de antecipação da entrega de um determinado pedido de venda pendente.

No entanto, para responder a essa pergunta, os analistas do departamento comercial precisam refazê-la para os analistas do PCP, que é o setor responsável pelo planejamento, programação e controle da produção.

A intenção do PCP consiste em equilibrar a capacidade produtiva à demanda sinalizada pelo mercado, fazendo com que a operação produza [e compre apenas] o necessário para o atendimento à demanda, sem excessos ou faltas, de forma a garantir que o nível de produção seja suficiente para atender as necessidades dos clientes (LOZADA, 2017, p. 83).

Já os programadores do setor de PCP, por sua vez, para responder essa pergunta com a qualidade, propriedade e segurança necessária para a área comercial, precisam efetuar uma análise complexa de capacidade *versus* demanda da fábrica, de modo a garantir que essa antecipação solicitada por um cliente específico não interfira na programação de entrega de todos os demais pedidos de venda que já foram firmados, inclusive do mesmo cliente.

Ou seja, dependendo da agilidade, disponibilidade, grau de conhecimento e experiência de todos os analistas envolvidos no processo, e do desempenho das ferramentas utilizadas para análise, a resposta para o cliente final pode demorar vários dias, sem ter a certeza de que sua demanda será efetivamente atendida.

Além disso, na maioria das vezes, todas as interações entre o cliente e o técnico comercial, e o técnico comercial e o programador de produção, são realizadas via correspondência eletrônica, fator que acaba onerando ainda mais o processo como um todo.

Ademais, é bem comum que para não desperdiçar muito tempo com essa atividade, o programador de produção renuncie a qualidade e a assiduidade que essa tarefa exige, e acaba realizando apenas uma análise superficial da solicitação comercial, sem avaliar todos os impactos inerentes ao aceite da nova data de entrega do pedido em questão.

O cenário inverso também é recorrente, isto é, por várias vezes, é o programador de produção que solicita ao técnico comercial a renegociação de um pedido do cliente com postergação da data de entrega, porque é notório que a produção não terá condições de entregar o pedido no prazo inicialmente acordado ou, pior ainda, porque o prazo já se esgotou e o item não foi entregue como prometido.

Consequentemente, de forma direta ou indireta, a organização pode se deparar diante ao enfrentamento de graves problemas que são inadmissíveis perante o competitivo mercado em que atua, conforme relacionado a seguir:

- a) muitos questionamentos de pedidos de venda com respostas incompletas ou inadequadas;
- b) descumprimento dos prazos de entrega prometidos aos clientes;
- c) incredulidade dos clientes atinente ao prazo de entrega estimado para seus pedidos;
- d) resultados indesejáveis nos indicadores de atendimento ao cliente;
- e) insatisfação dos clientes com o atendimento comercial, principalmente, no que diz respeito à agilidade no retorno de suas solicitações;
- f) atravessamento de muitas ordens de produção na fábrica, ignorando o planejamento e o sequenciamento que é cuidadosamente elaborado pelo planejador mestre do PCP;
- g) custos extras devido aos atrasos na entrega de pedidos de venda, como por exemplo, com fretes expressos e outras despesas não planejadas para transporte emergencial de mercadorias.

1.2 QUESTÃO DE PESQUISA

À vista disso, a questão de pesquisa que orienta este trabalho de conclusão de curso (TCC) é a seguinte: como agilizar e padronizar as respostas para as perguntas mais frequentes sobre os pedidos de venda de uma determinada indústria do setor metalomecânico?

1.3 OBJETIVOS

Esta seção apresenta todos os objetivos que se busca atingir ao final deste trabalho de conclusão de curso.

1.3.1 Objetivo geral

O objetivo geral deste trabalho consiste em desenvolver um *chatbot* que seja capaz de responder as perguntas mais frequentes sobre os pedidos de venda de uma determinada indústria do setor metalomecânico.

Todavia, para que seja possível responder ao usuário de uma forma rápida e precisa, o *chatbot* desenvolvido terá que ser integrado ao sistema *Drummer APS*, que já é utilizado pela organização em questão.

Além disso, é imprescindível respeitar todos os princípios de governança do setor de TI da empresa, bem como, a preferência por produtos, serviços e plataformas de desenvolvimento da *Microsoft*. Inclusive, o *chatbot* deve ser disponibilizado na ferramenta *Microsoft Teams*, pois esse é o canal oficial de comunicação interna do Grupo Sazi.

1.3.2 Objetivos específicos

De modo complementar ao objetivo geral, e com o propósito de atingi-lo da melhor forma possível, almeja-se alcançar os seguintes objetivos específicos neste trabalho:

- a) estudo dos principais conceitos de IA envolvidos no processo de desenvolvimento de *chatbots*;
- b) análise de técnicas de integração de sistemas que possam ser adequadas para solucionar o problema;
- c) levantamento e especificação dos requisitos do sistema;
- d) análise de soluções específicas da *Microsoft* para o desenvolvimento de *chatbots*;
- e) projeto, desenvolvimento e testes do *chatbot*;
- f) piloto e pesquisa qualitativa do *chatbot* com os usuários chave do projeto;
- g) avaliação dos resultados obtidos.

1.4 METODOLOGIA

Este trabalho de conclusão está fundamentado no método científico de estudo de caso com aplicação em organização.

É importante salientar que este projeto de diplomação foi dividido em duas fases, denominadas I e II, sendo a primeira fase de pesquisa, especificação e projeto, e a segunda fase de desenvolvimento, testes e avaliação.

Tendo ciência disso, e visando atingir todos os objetivos aqui propostos, as seguintes etapas foram planejadas para a fase I deste projeto:

- a) fundamentação teórica;
- b) definição de usuários chave;
- c) elicitação de requisitos;
- d) projeto de *software*.

Já a fase II do projeto, que inicia logo após o término da fase I, foi subdividida nas etapas listadas a seguir:

- e) desenvolvimento;
- f) testes;
- g) piloto;
- h) avaliação dos resultados.

As atividades de cada etapa do projeto, tanto da fase I, quanto da fase II, foram sintetizadas nas seções seguintes.

1.4.1 Fase I

Inicialmente, será realizado um estudo bibliográfico bem abrangente sobre os principais assuntos abordados neste projeto, com o objetivo de formar uma base de conhecimento sólida e satisfatória para a elaboração do referencial teórico deste TCC. Segue alguns dos temas que serão objetos de estudo e pesquisa neste trabalho:

- a) inteligência artificial;
- b) processamento de linguagem natural;
- c) *chatbots*;
- d) sistemas de informação;
- e) integração de sistemas.

Após finalizado o estudo, serão definidos os usuários chave do projeto, cujo papel principal será o de fornecer informações referente às áreas de negócio envolvidas, e terão participação ativa nas etapas de validação do sistema que será desenvolvido neste trabalho.

Logo na sequência, os requisitos funcionais e não funcionais do *chatbot* serão elicitados utilizando a técnica de entrevistas abertas com os usuários chave (SOMMERVILLE, 2018) e uma matriz SIPOC como ferramenta de apoio para mapear quais são as perguntas mais frequentes sobre pedidos de venda. Essa etapa também tem o objetivo de alinhar a expectativa dos usuários chave quanto ao produto final que será gerado por este projeto que possui apenas o foco experimental.

Imediatamente depois de esclarecidos todos os requisitos do projeto com os usuários chave, dá-se início a etapa de projeto de *software*, sendo a primeira atividade uma análise detalhada de soluções da *Microsoft* para o desenvolvimento de *chatbots*. Após a análise, será selecionada e apresentada a plataforma de desenvolvimento de *software* que viabilizará a construção de um *chatbot* integrado ao sistema *Drummer APS* e à ferramenta de comunicação *Microsoft Teams*. Também é nesse estágio que o *software* será projetado utilizando a Linguagem de Modelagem Unificada (UML).

Nesse ponto é importante enfatizar que somente alternativas para o desenvolvimento de *chatbots* oferecidas pela *Microsoft* foram estudadas e utilizadas neste trabalho, devido à preferência do setor de TI da empresa por produtos e serviços desse parceiro de negócios, e para garantir a manutenibilidade do *software* gerado, visto que já existe um conhecimento especializado da equipe nessa plataforma de soluções tecnológicas.

Assim, ao final dessa fase, espera-se determinar o escopo do projeto e construir todos os artefatos que são minimamente necessários para iniciar o processo de desenvolvimento do *chatbot*. Desse modo, uma execução bem-sucedida até aqui é imprescindível para garantir o êxito das etapas subsequentes e assegurar o sucesso de todo o projeto.

1.4.2 Fase II

Com base em todos os artefatos produzidos na etapa de projeto de *software*, a primeira atividade da fase II deste trabalho será desenvolver o *chatbot* utilizando as plataformas e as linguagens de programação definidas na fase predecessora.

Após concluída a etapa de desenvolvimento do produto, uma versão *alfa* do *chatbot* será submetida a execução de testes de funcionalidade, usabilidade, navegabilidade, estabilidade, integração, desempenho e segurança, pois essas características são primordiais para garantir um produto de qualidade, de acordo com a norma ABNT NBR ISO/IEC 9126-1:2003. Logo, a saída da etapa de testes do *chatbot* será a identificação e a correção de eventuais falhas do sistema, além da garantia de que todos os requisitos foram plenamente atendidos pelo produto gerado.

Dessa maneira, uma versão *closed beta* do *chatbot* será liberada ao grupo de usuários chave previamente definido para realização de um piloto do projeto em um ambiente de homologação controlado. Durante esse piloto, será utilizada a técnica de observação estruturada para coletar dados referente a aderência do produto para os processos aos quais o sistema se predispôs a apoiar (SOMMERVILLE, 2018).

Para concluir, a última atividade dessa fase será a realização de uma pesquisa qualitativa por meio de uma entrevista estruturada com os usuários chave a fim de avaliar os resultados do projeto e validar se os objetivos do trabalho foram plenamente atingidos.

1.5 ESTRUTURA DO TEXTO

O texto desta monografia foi estruturado em seis capítulos, considerando o capítulo atual.

Este capítulo introdutório tem como objetivo contextualizar o leitor a respeito da motivação e da relevância do assunto abordado neste trabalho, apresentar todos os objetivos que se pretende alcançar ao final deste projeto, revelar qual foi a metodologia adotada para desenvolver este trabalho, e esclarecer como o texto foi organizado.

O próximo capítulo apresenta os conceitos básicos e algumas áreas de aplicação da inteligência artificial e, em especial, introduz os *chatbots*.

O Capítulo 3 tem como objetivo apresentar brevemente a história e as características dos principais sistemas de informação e sintetizar as técnicas de integração de sistemas que foram estudadas para viabilizar o desenvolvimento deste trabalho.

O Capítulo 4 propõe uma solução para o problema de pesquisa abordado neste capítulo e se limita em descrever as atividades executadas na fase I deste projeto.

Já o Capítulo 5 detalha como a solução especificada no capítulo anterior foi implementada, compreendendo todas as etapas da fase II deste projeto.

Por fim, o Capítulo 6 conclui este trabalho apresentando as considerações finais do autor, expondo algumas lições aprendidas durante a execução deste projeto, avaliando os resultados obtidos com este TCC e recomendando trabalhos futuros.

2 INTELIGÊNCIA ARTIFICIAL

Inteligência artificial, derivado do termo em inglês *artificial intelligence*, e muitas vezes mencionada apenas pela sigla em português IA, ou pela sigla em inglês AI, é uma área de estudo da ciência da computação que possui a intenção de tornar os sistemas mais inteligentes e aderentes a este complexo mundo.

Para Luger (2004, p. 23), a “Inteligência artificial (IA) pode ser definida como o ramo da ciência da computação que se ocupa da automação do comportamento inteligente.”. Porém, o que é um comportamento inteligente?

Alan Mathison Turing, renomado matemático e cientista da computação britânico, que também é mundialmente reconhecido como sendo o pai da computação moderna, respondeu a essa pergunta em seu pioneiro artigo sobre inteligência artificial publicado em 1950 na revista *Mind*. A obra intitulada “*Computing Machinery and Intelligence*” define que uma máquina apresenta um comportamento inteligente quando alcança um elevado desempenho cognitivo, a ponto de conseguir enganar um ser humano que a interroga. Na seção seguinte será explicado esse importante conceito que foi batizado de Teste de Turing.

2.1 TESTE DE TURING

“O Teste de Turing mede o desempenho de uma máquina aparentemente inteligente, em relação ao desempenho de um ser humano, indiscutivelmente o melhor e único padrão para comportamento inteligente.” (LUGER, 2004, p. 31).

Na prática, o teste proposto por Turing em 1950 é uma variação do “Jogo da Imitação”, do inglês “*The Imitation Game*”. Nesse teste, três participantes, sendo dois humanos e um computador, são isolados em salas separadas, e um dos participantes humanos, denominado interrogador, é desafiado a determinar qual dos outros dois participantes é o computador. Sendo que, o interrogador pode utilizar apenas um terminal para comunicar-se textualmente com os demais participantes. Se o interrogador não for capaz de distinguir a máquina do ser humano que conversa, então o computador é aprovado no teste e, afirma Turing (1950), pode-se supor que a máquina em questão é inteligente e pode pensar.

Russell e Norvig (2004, p. 4) destacam que as seguintes capacidades eram necessárias para que um computador fosse aprovado no Teste de Turing:

processamento de linguagem natural, representação de conhecimento, raciocínio automatizado e aprendizado de máquina. Tais capacidades, até hoje, são extremamente relevantes na criação de mecanismos virtuais inteligentes e, conseqüentemente, na área de inteligência artificial.

2.2 ÁREAS DE APLICAÇÃO

Assim como ocorre na grande maioria dos campos de estudo da ciência da computação, a IA é bastante ampla e pode ser aplicada nas mais diversas áreas.

Esta seção apresenta uma visão geral de algumas das mais importantes áreas de aplicação da inteligência artificial, de acordo com Luger (2004, p. 38).

2.2.1 Jogos

Tendo a busca em espaço de estados como paradigma base, muitas pesquisas em IA foram desenvolvidas sobre jogos, principalmente de tabuleiro (xadrez, por exemplo), visto que a maioria desses jogos utiliza um conjunto de regras bem definido que evita várias das ambigüidades e sutilezas semânticas que são inerentes aos problemas de domínio mais complexo. Além disso, os jogos podem ser testados facilmente e sem ônus financeiro ou ético.

Porém, alguns jogos podem ser complexos o suficiente a ponto de exigirem técnicas avançadas, também chamadas de heurísticas, para determinais quais são as melhores alternativas a serem exploradas no espaço do problema.

Citando Luger (2004, p. 39), “Muito do que chamamos de inteligência parece estar relacionado com as heurísticas usadas pelos seres humanos para resolver problemas.”.

2.2.2 Raciocínio automático

Sendo considerado o ramo mais antigo da inteligência artificial, “a prova automática de teoremas foi responsável por muitos trabalhos iniciais na formalização de algoritmos de busca e no desenvolvimento de linguagens de representação formais” (LUGER, 2004, p. 39).

A lógica é um sistema formal e de raciocínio matemático que pode ser automatizada e utilizada para solucionar uma ampla gama de problemas dispostos como teoremas a serem provados.

Infelizmente, nem todos os problemas podem ser solucionados por um simples sistema lógico, principalmente os problemas mais complicados, devido ao fato de que podem existir um número infinito de teoremas prováveis, que sem as heurísticas mais adequadas para guiar a busca, fazem com que os provedores automáticos de teoremas demorem muito até encontrar o teorema correto. Por causa disso, muitos pesquisadores afirmam que a única alternativa é confiar em estratégias informais que estão ligadas ao desenvolvimento de sistemas especialistas.

2.2.3 Sistemas especialistas

Como afirma Luger (2004, p. 40), “O conhecimento especialista é uma combinação de um entendimento teórico do problema com uma coleção de regras heurísticas para resolver problemas, que a experiência demonstrou ser efetiva no domínio.”. Ou seja, o conhecimento específico de um determinado tema é a habilidade de alguém que aplica o conhecimento teórico e empírico sobre o domínio em questão.

Sabendo disso, podemos dizer que um sistema especialista é construído com base no conhecimento de um especialista humano no domínio, para que o computador seja capaz de resolver problemas bem específicos de uma determinada área.

Além do programa ser codificado com a colaboração do conhecimento de um especialista no segmento, o mesmo deve ser testado e refinado sob a supervisão crítica do mesmo especialista. De modo que, o programa atinja um elevado nível de desempenho e torne-se artificialmente inteligente.

Sistemas especialistas são implementados para resolver problemas das mais diversas áreas do conhecimento, como por exemplo, da computação, medicina, química, geologia, educação, negócios e ciência.

Entretanto, para que um programa especialista seja eficaz, é necessário que o domínio alvo já possua estratégias de solução para os problemas bem definidas, pois os problemas que dependem do senso comum são muito mais difíceis de serem resolvidos por esse método.

Luger (2004, p. 41) lista as seguintes deficiências e limitações como sendo as mais comuns dos sistemas especialistas: dificuldade em capturar o conhecimento profundo do domínio do problema; falta de robustez e flexibilidade; incapacidade em fornecer explicações profundas sobre o problema; árdua homologação; baixos níveis de aprendizagem por experiência.

2.2.4 Compreensão da linguagem natural

Desenvolver sistemas capazes de compreender a linguagem humana e se expressar de forma natural sempre foi um dos principais objetivos da inteligência artificial. Contudo, Luger (2004, p. 42) salienta que:

Compreender a linguagem natural envolve muito mais do que simplesmente analisar sentenças, separando-as em suas partes individuais, e procurar estas palavras num dicionário. A compreensão real depende de um extenso conhecimento do domínio do discurso e das expressões idiomáticas utilizadas naquele domínio, bem como da habilidade em aplicar conhecimento contextual genérico para resolver omissões e ambiguidades que são parte usual da fala humana.

Desse modo, fica claro que compreender a linguagem natural não é uma tarefa fácil, mas os pesquisadores já desenvolveram várias técnicas diferentes para estruturar o significado semântico usado no campo da inteligência artificial.

2.2.5 Modelagem do desempenho humano

Alguns sistemas de IA modelam explicitamente aspectos do desempenho humano, e possuem várias similaridades com a arquitetura da mente humana. Em virtude disso, muitos psicólogos adotaram a computação para formular modelos e testar teorias da cognição humana (LUGER, 2004, p. 43).

2.2.6 Planejamento e robótica

A pesquisa na área de planejamento surgiu devido a necessidade de projetar robôs considerados inteligentes que tivessem a habilidade de executar ações atômicas com um elevado grau de flexibilidade e sensibilidade em relação ao mundo exterior.

“Uma tarefa aparentemente simples, como decompor um problema em subproblemas independentes, na verdade requer heurísticas sofisticadas e conhecimento extensivo sobre o domínio do planejamento.” (LUGER, 2004, p. 44).

Uma das técnicas de planejamento utilizadas em robótica é a solução de problemas orientada a agentes, sendo que “Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores.” (RUSSELL; NORVIG, 2004, p. 33).

2.2.7 Linguagens e ambientes para IA

A pesquisa em IA também trouxe avanços significativos em linguagens de programação e ambientes de desenvolvimento de *software*.

Segundo Luger (2004, p. 45), esses “ambientes de programação incluem técnicas de estruturação do conhecimento, como a programação orientada a objetos, e estruturas para construção de sistemas especialistas”. Ademais, algumas linguagens de programação de alto nível, como LISP e PROLOG, foram desenvolvidas para aliviar a complexidade da busca guiada por heurísticas.

2.2.8 Aprendizagem de máquina

Talvez uma das subáreas mais importantes e desafiadoras da IA seja a aprendizagem de máquina, ou *Machine Learning* (ML).

A maioria dos sistemas especialistas é prejudicada pela inflexibilidade das suas estratégias para resolução de problemas e pela dificuldade em modificar grandes quantidades de código. A solução óbvia para estes problemas é permitir que estes sistemas aprendam por conta própria, quer pela sua própria experiência, por analogia ou por exemplos, ou, então, por um professor que lhes “diga” o que fazer. (LUGER, 2004, p. 45).

2.2.9 Redes neurais e algoritmos genéticos

Uma outra abordagem de IA, com maior potencial para reconhecer dados ruidosos e incompletos, é através do desenvolvimento de sistemas “que imitam a estrutura dos neurônios no cérebro humano, ou os padrões evolutivos encontrados nos algoritmos genéticos e em vida artificial” (LUGER, 2004, p. 46).

Luger (2004, p. 47) descreve uma rede neural como um modelo computacional de processamento paralelo e distribuído, em que cada unidade calcula uma função específica com base em suas entradas e transfere o resultado para outras unidades que estão conectadas a ela na mesma rede.

2.2.10 Filosofia

A IA possui uma enorme contribuição filosófica no que tange ao entendimento da própria inteligência, e também explora algumas questões bem profundas, como por exemplo, aquela proposta por Alan Turing: “As máquinas podem pensar?” (TURING, 1950, p. 433, tradução nossa).

2.3 CHATBOTS

Um exemplo de aplicação contemporânea da inteligência artificial são os *chatbots*, porque eles são sistemas especialistas que compreendem a linguagem natural do ser humano (PLN) e aprendem mais sobre o usuário e suas necessidades por meio de algoritmos avançados de aprendizagem de máquina (ML). Além do mais, há também alguns *chatbots* que implementam redes neurais artificiais para criar uma experiência ainda mais humana para quem os utiliza. Ou seja, os *chatbots* são extremamente flexíveis e podem englobar inúmeras subáreas da IA.

A palavra *chatbot* é uma abreviação do termo *chatterbot*, que surgiu por meio da junção de duas palavras de origem inglesa: *chatter*, que pode ser interpretado como um indivíduo tagarela, e *bot*, que é uma abreviação da palavra *robot* que significa robô em português. Ou seja, literalmente, a expressão *chatterbot* poderia ser traduzida como um “robô tagarela”.

Por essa razão, o termo *chatbot* é utilizado para se referenciar a um sistema de conversação que interage com os usuários via linguagem natural (SHAWAR; ATWELL, 2005, p. 489, tradução nossa).

Ainda, segundo Reshmi e Balakrishnan (2016, p. 1173, tradução nossa), um *chatbot* pode ser classificado como uma entidade artificial designada a simular uma conversa inteligente com seres humanos, respondendo por meio de linguagem natural, e de modo a imitar uma pessoa real.

2.3.1 História dos chatbots

O termo *chatterbot* foi cunhado em 1994 por Michael Loren Mauldin, cientista da *Carnegie Mellon University*, quando apresentou o seu primeiro *chatbot*, que foi chamado de Julia, na *Twelfth National Conference on Artificial Intelligence* (MAULDIN, 1994). Porém, programas de computador desse tipo já existiam há muito mais tempo.

Para entender a origem desses agentes conversacionais, é preciso voltar no tempo para o ano de 1940, quando o famoso escritor e bioquímico russo, Isaac Asimov, começa a redigir uma série de nove contos sob o conceito disruptivo de robôs inteligentes que eram capazes de conversar e pensar. Até que, finalmente, em 1950, Asimov reuniu toda essa coletânea de contos em uma obra literária marcante de ficção científica que foi intitulada “Eu, Robô”, que apresentou as três leis fundamentais da robótica (ASIMOV, 1969; CRUZ; ALENCAR; SCHMITZ, 2018).

No mesmo ano de 1950, é inevitável recordar que o cientista da computação Alan Turing se antecipou no tema da existência de mecanismos virtuais inteligentes, quando propôs um teste para determinar se uma máquina podia enganar um ser humano que a interroga (CRUZ; ALENCAR; SCHMITZ, 2018; TURING, 1950).

Dessa forma, pode-se afirmar que esses dois fatos históricos ocorridos no ano de 1950, tanto o lançamento do livro “Eu, Robô” de Isaac Asimov, quanto o surgimento do Teste de Turing, serviram como base e fonte de inspiração para os próximos acontecimentos relacionados à inteligência artificial e, mais especificamente, ao desenvolvimento de *chatbots* (CRUZ; ALENCAR; SCHMITZ, 2018).

Em 1966, o teuto-estadunidense Joseph Weizenbaum, pesquisador e professor emérito do famoso Instituto de Tecnologia de Massachusetts (MIT), desenvolveu o primeiro *chatbot* do mundo: ELIZA. O *software* ELIZA ficou conhecido dessa forma, porque foi o primeiro algoritmo de processamento de linguagem natural implementado da história (CRUZ; ALENCAR; SCHMITZ, 2018; WEIZENBAUM, 1966).

Com o objetivo de passar no Teste de Turing, esse robô de conversação chegou bem perto de imitar um ser humano, durante a simulação de um diálogo entre um usuário paciente e sua psicoterapeuta virtual da linha rogeriana, cuja abordagem é centrada no indivíduo. Resumidamente, a técnica utilizada pelo programa ELIZA era relativamente simples: após a entrada de uma determinada frase pelo usuário, o programa identificava as palavras-chave da frase e as comparava com um conjunto

de regras pré-programadas para gerar a resposta mais adequada ao usuário (WEIZENBAUM, 1966).

Desde a ELIZA, houve um enorme progresso no desenvolvimento de *chatbots*. Apenas alguns anos mais tarde, em 1972, Kenneth Colby, renomado psiquiatra americano e cientista de Stanford, aperfeiçoa o ELIZA e programa o robô de conversação PARRY, que simula um paciente diagnosticado com esquizofrenia paranoide, um tipo de distúrbio mental grave. O objetivo do professor era criar uma ferramenta de ensino com realidade virtual efetiva para treinamento dos estudantes de psiquiatria (COLBY, 1981; CRUZ; ALENCAR; SCHMITZ, 2018).

Somente doze anos depois, em 1984, surge o primeiro *chatbot* brasileiro: Carla. Inspirado em ELIZA, um garoto de apenas quatorze anos, Rodrigo de Almeida Siqueira, programou a psicanalista virtual Carla, cujo estudo foi publicado na edição número 12 da revista *Microhobby* (RENZETTI, 1984).

Já na década de 90, em 1995, o Dr. Richard Wallace, da *Lehigh University*, desenvolveu o *chatbot* ALICE, abreviatura de *Artificial Linguistic Internet Computer Entity*, um dos mais famosos da atualidade. Significativamente mais complexo que seus antecessores, esse sistema foi precursor no uso da linguagem de marcação AIML (*Artificial Intelligence Markup Language*), uma extensão do XML (*Extensible Markup Language*), que ainda é muito utilizada (WALLACE, 2009).

Desde então, vislumbrando o enorme potencial que essa tecnologia emergente apresentava, a maioria das grandes empresas do ramo tecnológico começaram a investir no desenvolvimento de agentes de conversação cada vez mais avançados e inteligentes. Tão grande foi a evolução dos *chatbots*, que o mercado passou a utilizar o sinônimo de assistente virtual inteligente para denominá-los, conforme segue alguns exemplos: *Watson* da IBM, *Siri* da Apple, *Now* da Google, *Cortana* da Microsoft e *Alexa* da Amazon.

2.3.2 Atributos dos chatbots

Como existe uma variedade muito grande de *chatbots* disponíveis no mercado, Cruz, Alencar e Schmitz (2018) classificam os *chatbots* pelos seus atributos, conforme resumido no Quadro 1 e detalhado a seguir:

a) **comportamento:**

- passivos: ativados apenas quando o usuário solicita auxílio;
- dinâmicos: ativados automaticamente quando o usuário aparece;
- dinâmicos com gatilho: ativados automaticamente quando o usuário executa alguma ação específica que demonstre a necessidade de auxílio;

b) **propósito:**

- gerais: interagem com o usuário sobre qualquer assunto;
- especializados: auxiliam o usuário sobre assuntos específicos;

c) **apresentação:**

- com avatar: personificados na forma de uma figura humana ou robô;
- sem avatar: completamente abstratos e sem personalidade;

d) **comunicação:**

- sociáveis: atenciosos, educados e preocupados com a qualidade das informações que fornece;
- indiferentes: mecânicos, apáticos e despreocupados com a qualidade das informações que fornece;

e) **memória:**

- com memória: lembram das informações fornecidas pelo usuário durante a conversa e as utiliza para facilitar e agilizar a comunicação;
- sem memória: respondem as perguntas do usuário sempre da mesma maneira, independentemente do contexto;

f) **idioma:**

- monoglotas: comunicam-se em uma língua única e específica;
- políglotas: comunicam-se em várias línguas;

g) **expressões idiomáticas:**

- flexíveis: capazes de lidar com abreviaturas, siglas, sinônimos, gírias, jargões, entre outras expressões idiomáticas;

- inflexíveis: precisam que as questões sejam apresentadas em um formato específico, livre de expressões idiomáticas;
- h) aprendizado:**
- com capacidade: aumentam automaticamente a sua habilidade de interagir com o usuário ao longo do tempo;
 - sem capacidade: independentemente da quantidade de interações com o usuário, sua habilidade permanece sempre a mesma;
- i) formato de apresentação das informações:**
- multimídia: utilizam texto com imagem, áudio ou vídeo;
 - texto: utilizam apenas texto;
- j) reposicionamento de página da internet:**
- com capacidade: capazes de reposicionar a página que o usuário está observando;
 - sem capacidade: incapazes de reposicionar a página que o usuário está observando;
- k) sugestões de consulta a tópicos de interesse:**
- proativos: sugerem a consulta de tópicos adicionais que possam ser de interesse do usuário;
 - reativos: limitam-se a responder apenas as perguntas do usuário;
- l) integração com sistemas de informação:**
- integrados: acessam sistemas de informação corporativos para consultar ou atualizar dados solicitados pelo usuário;
 - não integrados: não possuem acesso aos sistemas de informação corporativos e limitam-se apenas às informações que foram programados para fornecer.

Obviamente, que “[...] quanto mais sofisticado for o assistente virtual, mais recursos ele irá consumir para ser construído e mantido ao longo do tempo.” (CRUZ; ALENCAR; SCHMITZ, 2018, p. 42).

Quadro 1 – Atributos para classificação de *chatbots*

#	Atributo	Sim/Não
A	Comportamento	<input type="checkbox"/> Passivo
		<input type="checkbox"/> Dinâmico
		<input type="checkbox"/> Dinâmico c/ gatilho
B	Propósito	<input type="checkbox"/> Geral
		<input type="checkbox"/> Especializado
C	Apresentação	<input type="checkbox"/> Com avatar
		<input type="checkbox"/> Sem avatar
D	Forma da Comunicação	<input type="checkbox"/> Sociável
		<input type="checkbox"/> Indiferente
E	Memória	<input type="checkbox"/> Com memória
		<input type="checkbox"/> Sem memória
F	Idioma	<input type="checkbox"/> Monoglota
		<input type="checkbox"/> Poliglota
G	Expressões idiomáticas	<input type="checkbox"/> Flexíveis
		<input type="checkbox"/> Inflexíveis
H	Aprendizado	<input type="checkbox"/> Com capacidade
		<input type="checkbox"/> Sem capacidade
I	Formato de apresentação das informações	<input type="checkbox"/> Multimídia
		<input type="checkbox"/> Texto
J	Reposicionamento de página da Internet	<input type="checkbox"/> Com capacidade
		<input type="checkbox"/> Sem capacidade
K	Sugestões de consulta a tópicos de interesse	<input type="checkbox"/> Proativo
		<input type="checkbox"/> Reativo
L	Integração com sistemas de informação	<input type="checkbox"/> Integrado
		<input type="checkbox"/> Não integrado

Fonte: Cruz, Alencar e Schmitz (2018, p. 41).

2.3.3 Processamento de linguagem natural

A grande maioria dos *chatbots* desenvolvidos pelo mercado utilizam algoritmos de processamento de linguagem natural (PLN), do inglês *Natural Language Processing* (NLP), para serem capazes de se comunicar com os usuários. Basicamente, o PLN é uma subárea da IA responsável por interpretar a língua dos seres humanos (CRUZ; ALENCAR; SCHMITZ, 2018).

Porém, essa tarefa não é tão simples quanto parece, visto que muito mais do que entender o significado literal de cada palavra que foi digitada pelo usuário, o *chatbot* precisa entender o contexto da conversa e compreender o significado sintático e semântico do texto (IBAÑOS; PAIL, 2015).

Segundo Ibaños e Pail (2015), outro recurso linguístico indispensável para qualquer sistema de PLN é o léxico computacional, que são as estruturas que

armazenam os itens lexicais associados a um conjunto de traços morfológicos, sintáticos, semânticos e pragmático-discursivos.

Normalmente, os algoritmos de PLN processam o texto recebido em forma de mensagem pelo usuário, e retornam para o programador no seguinte formato:

- a) **enunciado**: a mensagem integral do usuário, tal qual, ela foi digitada;
- b) **intenção**: é o desejo do usuário com o enunciado;
- c) **entidade**: qualquer dado do enunciado que complemente a intenção.

Por exemplo, se o usuário enviar a mensagem “Qual é a data de entrega do pedido 1?” para o *chatbot*, o enunciado será “Qual é a data de entrega do pedido 1?”, a intenção poderia ser “ConsultarPedido”, e uma entidade possível seria o número do pedido.

2.4 CONSIDERAÇÕES FINAIS

Graças ao estudo que foi realizado e brevemente apresentado neste capítulo, foi possível identificar que o *chatbot* desenvolvido por este trabalho deverá ser:

- a) passivo;
- b) especializado;
- c) sem avatar;
- d) sociável;
- e) com memória;
- f) monoglota;
- g) flexível;
- h) com capacidade de aprendizado;
- i) texto;
- j) sem capacidade de reposicionamento de página;
- k) reativo;
- l) integrado.

3 INTEGRAÇÃO DE SISTEMAS

Além de desenvolver um *chatbot*, este trabalho de conclusão tem como objetivo integrá-lo a um dos principais sistemas de informação utilizado pela empresa Máquinas Sazi, sendo que essa integração é fundamental para oferecer a melhor experiência possível para os usuários.

Ainda, de acordo com De Sordi e Marinho (2007), as organizações já reconhecem a importância da integração entre os sistemas e a consideram como sendo um componente crítico para o bom desempenho dos negócios.

Dessa forma, neste capítulo será apresentado e contextualizado o sistema de informação integrado ao *chatbot* desenvolvido por este trabalho, além das principais técnicas de integração que são utilizadas pelos desenvolvedores na atualidade.

3.1 SISTEMAS DE INFORMAÇÃO

Tecnicamente, um sistema de informação, também denominado SI, é um conjunto de componentes relacionados entre si que são responsáveis por coletar, processar, transformar, armazenar e distribuir informações com o objetivo de apoiar uma organização na tomada de decisões (LAUDON, 2015).

Todavia, é importante compreender o que quer dizer informação. Informação são dados brutos analisados e organizados em um formato acessível, inteligível e útil para a pessoa que a consome (LAUDON, 2015).

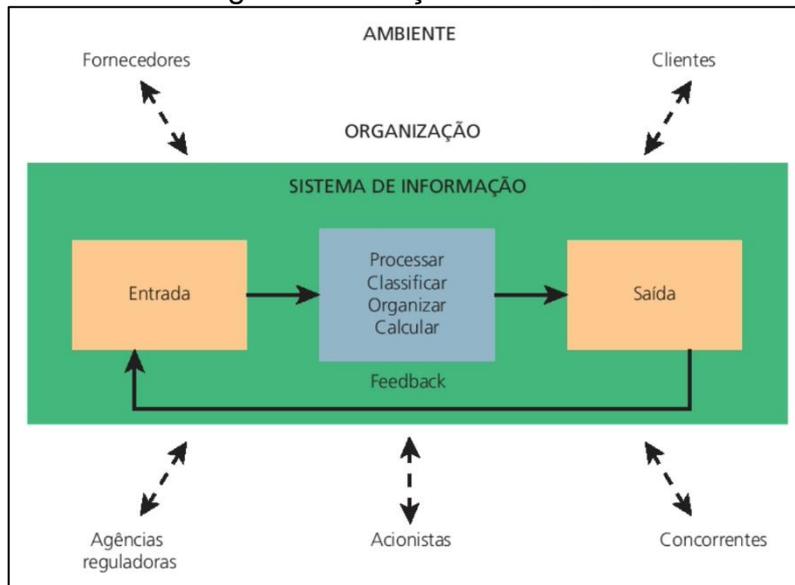
Isto posto, conforme Laudon (2015), um SI precisa executar três atividades essenciais para ser capaz de gerar as informações que as organizações precisam:

- a) **entrada:** coletar dados brutos do ambiente que cerca a organização;
- b) **processamento:** processar e transformar os dados brutos coletados em informações significativas;
- c) **saída:** distribuir as informações processadas aos usuários interessados.

Além dessas atividades, também é requerido um *feedback* da saída a alguns membros ou áreas da organização para posterior avaliação e constante refinamento da entrada (LAUDON, 2015).

A Figura 1 ilustra o fluxo básico de um sistema de informação.

Figura 1 – Funções de um SI



Fonte: Laudon (2015, p. 14).

Conforme expresso na Figura 2, outro detalhe interessante é que os sistemas de informação vão muito além dos computadores, e são formados por essas três dimensões harmoniosas que possuem o “[...] poder de fornecer soluções para os desafios e problemas no ambiente empresarial” (LAUDON, 2015, p. 15):

- a) **organizacional:** a história, a cultura, a estrutura, as funções, as premissas, os valores, as políticas e os processos de negócios inerentes às organizações são fatores determinantes na construção de um sistema de informação;
- b) **humana:** pessoas comprometidas, capacitadas e que saibam tirar o máximo proveito das informações geradas pelos sistemas são indispensáveis para o sucesso de qualquer organização. Essa é uma verdade que tem ainda maior impacto para os colaboradores dos níveis administrativos, porque a tecnologia da informação pode e deve ser utilizada como aliada para a manutenção, inovação e impulso dos negócios. Além, é claro, de que a eficácia de um SI está diretamente relacionada à qualidade técnica da equipe que o desenvolve e mantém;
- c) **tecnológica:** a infraestrutura de TI provê o alicerce para o perfeito funcionamento de todos os sistemas de informação da empresa. Nesse propósito, diversos recursos tecnológicos são administrados pela área de infraestrutura de TI, dentre eles: *hardware*, *software*, sistemas de gerenciamento de banco de dados (SGBD), redes de comunicações, *etc.*

Figura 2 – Dimensões de um SI



Fonte: Laudon (2015, p. 15).

Nas seções seguintes serão apresentados os sistemas de informação que serão integrados ao *chatbot* desenvolvido por este trabalho.

3.1.1 Sistema MRP

O sistema MRP, ou *Material Requirements Planning*, que significa planejamento das necessidades de materiais, surgiu na década de 1960 com o objetivo de calcular em que momento um determinado tipo e quantidade de material é necessário à operação para atender todas as demandas dos clientes no prazo (CORRÊA; GIANESI; CAON, 2019; LOZADA, 2017; TUBINO, 2017).

Para tanto, segundo Corrêa, Gianesi e Caon (2019), o MRP implementa a lógica de programação para trás, também chamada de *backward scheduling*, porque a partir da data em que o produto precisa ser entregue ao cliente, o sistema volta para trás no tempo explodindo todas as necessidades nível a nível. Obviamente, que o cálculo não é tão simples, pois precisa considerar uma série de parâmetros, tais como:

- a) a estrutura do produto;
- b) o roteiro de fabricação dos itens;
- c) os tempos de preparação e execução de cada operação;
- d) o tempo de reposição dos componentes comprados;
- e) as políticas de lotes mínimos, múltiplos e máximos;
- f) os tempos de segurança;
- g) os estoques de segurança;
- h) a posição de estoques.

Ademais, o cálculo é realizado com a premissa de disponibilizar os materiais na data exata em que são necessários, isto é, no último momento possível, visando reduzir os custos com a estocagem de produtos acabados ou intermediários (CORRÊA; GIANESI; CAON, 2019; LOZADA, 2017).

Contudo, apesar de ainda ser amplamente utilizado pelas empresas, o MRP possui severas restrições, como por exemplo, o fato de não considerar a capacidade produtiva no cálculo das necessidades. Por consequência, a empresa acaba arcando com custos adicionais, seja qual for a estratégia adotada para cobrir essa falha e viabilizar o cumprimento dos prazos de entrega. Algumas das alternativas utilizadas pelas fábricas para contornar esse problema são: trabalhar com capacidade em excesso, superestimar o *lead time* dos itens ou aumentar os estoques de segurança (CORRÊA; GIANESI; CAON, 2019).

3.1.2 Sistema MRP II

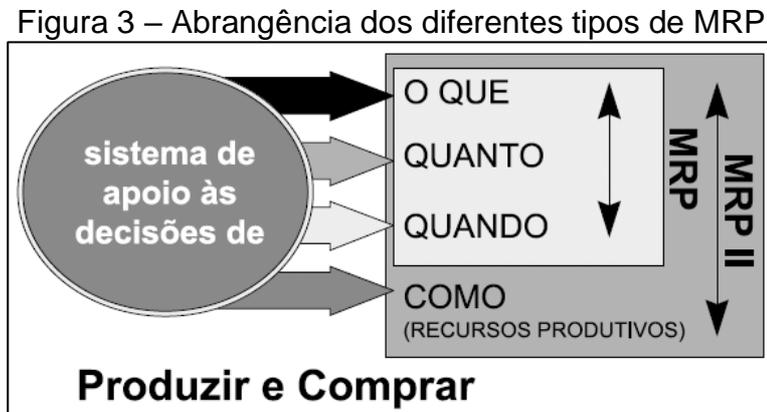
Alguns anos mais tarde, a partir de 1980, o MRP evoluiu consideravelmente para o chamado MRP II. Apesar de muitos acharem que a sigla MRP tenha o mesmo significado nos dois sistemas, na verdade, tratam-se de significados distintos. Enquanto MRP significa *Material Requirements Planning*, ou planejamento das necessidades de materiais, MRP II é um acrônimo para *Manufacturing Resource Planning*, ou planejamento dos recursos de manufatura (CORRÊA; GIANESI; CAON, 2019; LOZADA, 2017; TUBINO, 2017).

Como a própria tradução dos termos deixa transparecer, o MRP II é mais abrangente que seu predecessor, e possui a digna proposta de resolver os problemas do sistema anterior.

O MRP II considera a capacidade dos recursos produtivos em seu planejamento, sejam eles equipamentos ou mão de obra. Sendo assim, pode-se dizer que o MRP II calcula as necessidades de todos os recursos do processo de manufatura (CORRÊA; GIANESI; CAON, 2019; LOZADA, 2017; TUBINO, 2017).

Conforme ilustrado por Corrêa, Giansesi e Caon (2019) na Figura 3, o MRP II diferencia-se do MRP por orientar como os itens devem ser produzidos e comprados, em outras palavras, quais recursos produtivos devem ser acionados. Por essa razão, que Corrêa, Giansesi e Caon (2019) também afirmam que “[...] o novo cálculo [do MRP

II] permite reduzir custos com capacidade ociosa e estoques em excesso, mantendo os níveis de confiabilidade de entrega.”.



Fonte: Corrêa, Giansesi e Caon (2019, p. 140).

3.1.3 Sistema ERP

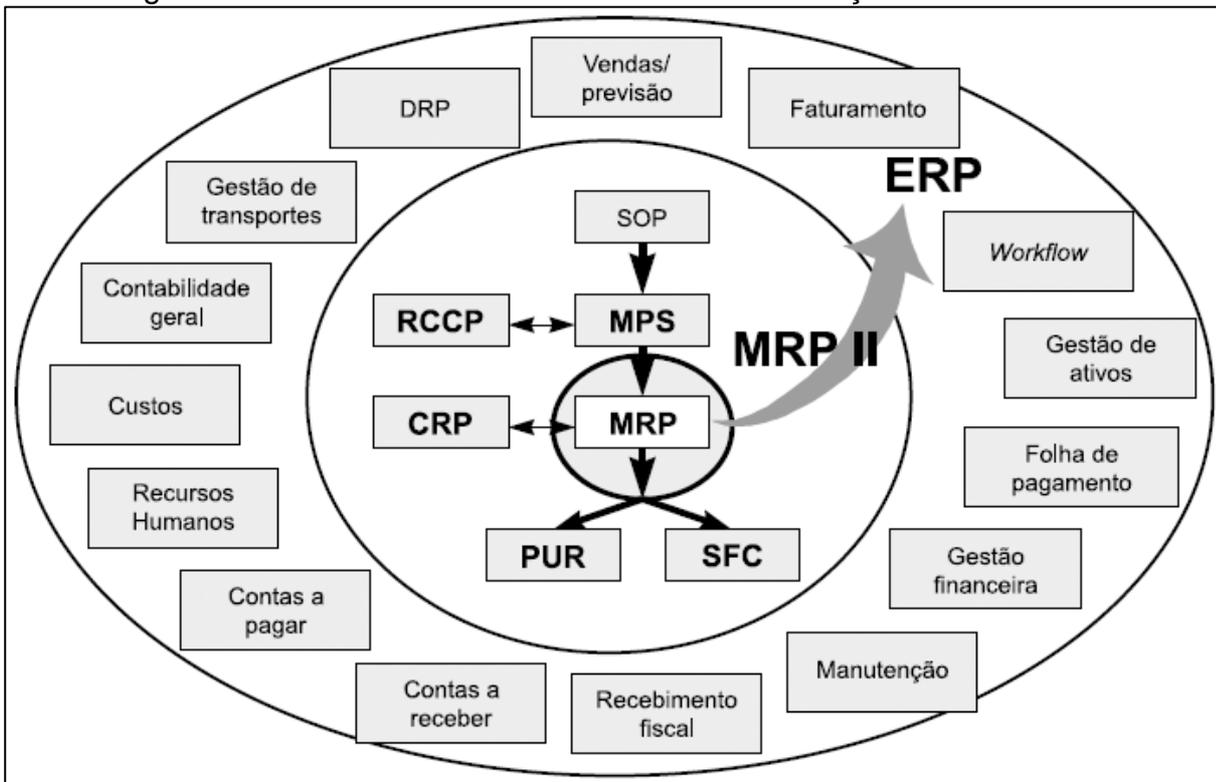
Já em meados da década de 1990, o MRP II deu origem ao ERP, que é a sigla para *Enterprise Resource Planning*, ou planejamento de recursos empresariais em tradução livre. No entanto, diferentemente do MRP II que é focado apenas na área de manufatura, o ERP é um sistema integrado de gestão que tem a finalidade de apoiar todos os setores da empresa na tomada de decisões (CORRÊA; GIANESI; CAON, 2019). Desse modo, é evidente que a abrangência dos sistemas ERP é muito maior do que a dos sistemas do tipo MRP II, conforme elucidado na Figura 4.

Didaticamente, o ERP pode ser definido “[...] como um sistema de informação adquirido na forma de pacotes comerciais de *software* que permitem a integração entre dados dos sistemas de informação transacionais e dos processos de negócios de uma organização.” (CAIÇARA, 2015, p. 96).

Nesse contexto, de acordo com Caiçara (2015), para que um sistema seja considerado um ERP é indispensável que apresente as seguintes características:

- a) ser adquirido pronto de um fornecedor especializado como um pacote comercial de *software*;
- b) ser construído com base nas melhores práticas de mercado;
- c) não ser desenvolvido para atender as necessidades de uma única empresa;
- d) ser modularizado;
- e) utilizar um banco de dados único e corporativo.

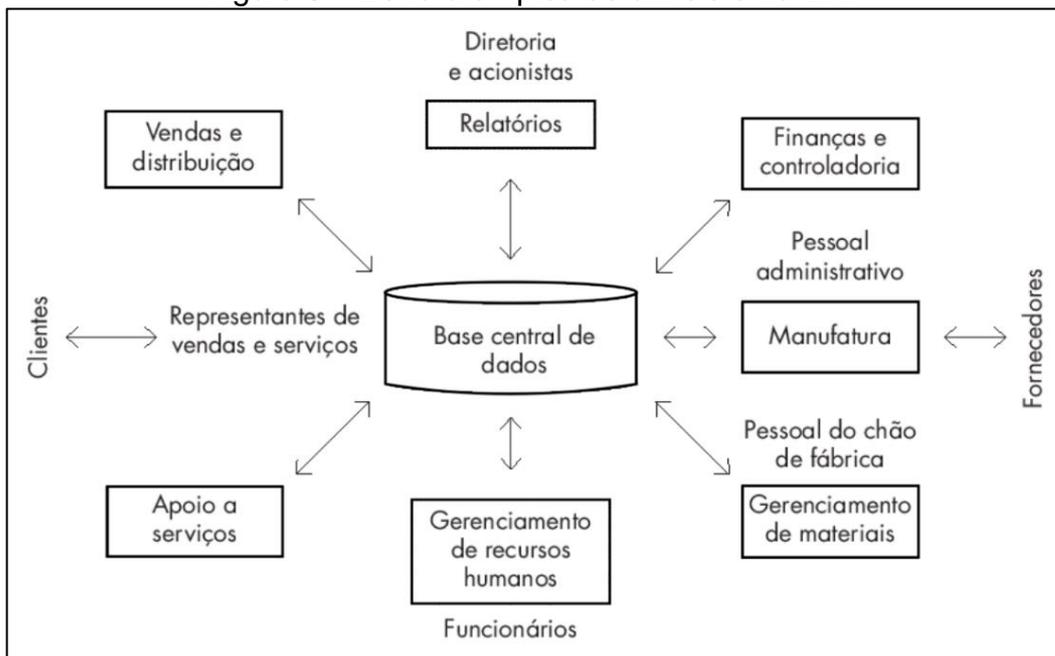
Figura 4 – Estrutura conceitual do ERP e sua evolução desde o MRP



Fonte: Corrêa, Giansesi e Caon (2019, p. 417).

A Figura 5 exibe a estrutura típica de funcionamento de um sistema ERP, que é composta por diversos módulos integrados entre si, a partir de uma base de dados única (CORRÊA; GIANESI; CAON, 2019).

Figura 5 – Estrutura típica de um sistema ERP



Fonte: Davenport (1998 apud CAIÇARA, 2015, p. 98).

Tão logo, as soluções caseiras desenvolvidas internamente por muitas empresas não podem ser classificadas nessa categoria, porque mesmo que esses produtos possuam várias semelhanças com um sistema ERP, inevitavelmente, eles são excessivamente customizados para atender aos processos específicos de uma única organização em um dado momento de sua existência (CAIÇARA, 2015, p. 97).

3.1.4 Sistema APS

Nos últimos anos, surgiram os mais diversos sistemas especialistas no mercado que visam trazer vantagens competitivas para as organizações que os adotam. Um exemplo de sistema especialista é o APS, ou *Advanced Planning and Scheduling*, que é um sistema de planejamento avançado de produção.

Segundo Tubino (2017), o APS também pode ser considerado um sistema de programação de capacidade finita, porque opera sobre um calendário real de disponibilidade dos recursos produtivos, sequenciando ordem a ordem, até o limite de tempo disponível por recurso.

Normalmente, os sistemas APS são integrados aos tradicionais sistemas ERP para evitar a duplicação de cadastros, mas eles também podem ser utilizados de forma independente de um ERP corporativo (TUBINO, 2017).

Esse tipo de sistema é capaz de executar o cálculo de MRP II para programar o que, quanto, onde, como e quando produzir ou comprar, evitando desperdícios e garantindo um melhor aproveitamento dos recursos disponíveis (LOZADA, 2017; MOELLMANN, 2017).

Contudo, os sistemas APS vão muito além disso, porque eles são responsáveis pelo planejamento da demanda, da produção, dos materiais (MRP), da capacidade dos recursos produtivos (CRP), da cadeia de suprimentos, dos estoques e da distribuição (DRP) (SIMCHI-LEVI *et al.*, 2003 apud MOELLMANN, 2017).

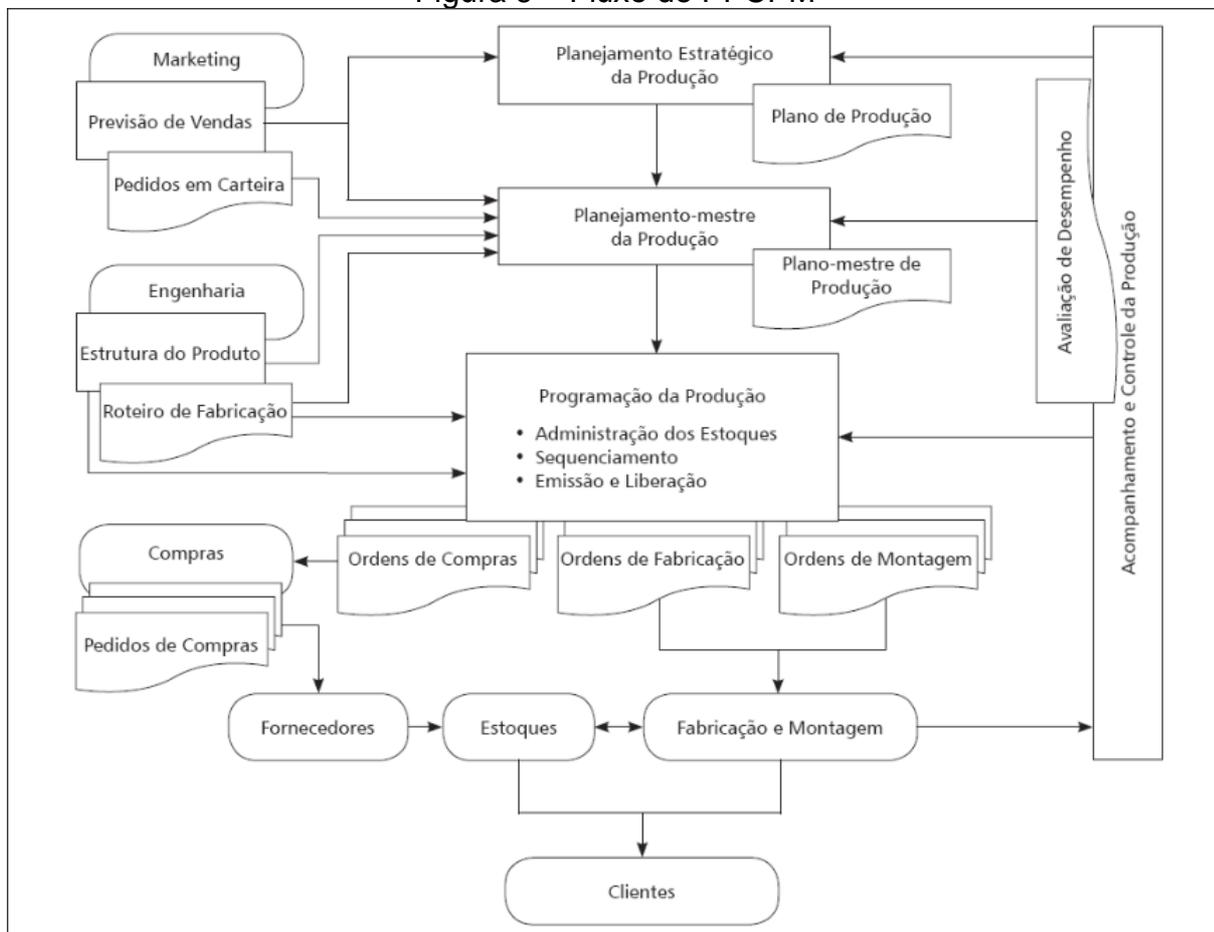
Nesse contexto, Corrêa, Gianesi e Caon (2019, p. 18) destacam que: “Planejar é entender como a consideração conjunta da situação presente e da visão de futuro influencia as decisões tomadas no presente para que se atinjam determinados objetivos no futuro.”.

Uma das principais funções de um *software* APS é o sequenciamento das ordens de compra (OC), ordens de fabricação (OF) e ordens de montagem (OM) planejadas pelo cálculo de MRP (TUBINO, 2017).

Bezerra (2014) destaca que a etapa de sequenciamento determina a ordem de execução das tarefas nos processos produtivos. Lozada (2017, p. 85) complementa que o processo de sequenciamento “[...] visa estabelecer a ordem de alocação das operações aos recursos, tendo como objetivos básicos a minimização dos tempos de processamento e dos atrasos.”.

Dessa forma, um *software* APS fornece suporte para quase todos os processos relacionados à área de Planejamento, Programação e Controle da Produção e dos Materiais (PPCPM), conforme fluxograma apresentado na Figura 6.

Figura 6 – Fluxo de PPCPM

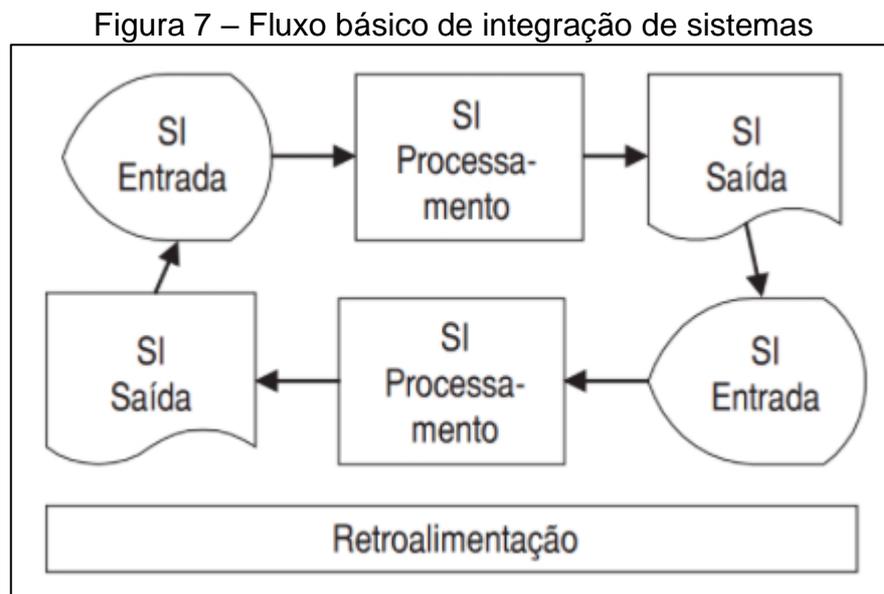


Fonte: Tubino (2017, p. 5).

3.2 TÉCNICAS DE INTEGRAÇÃO

Atualmente, são raros os sistemas que funcionam de forma completamente isolada e independente dentro das organizações. Da mesma forma que a integração entre os setores é fundamental para o sucesso de uma empresa, a integração de seus sistemas de informação proporciona enormes vantagens competitivas para o negócio (REZENDE; ABREU, 2013).

Para Rezende e Abreu (2013), as relações de interdependência entre os sistemas de informação são necessárias para o funcionamento efetivo das funções dentro de um ambiente empresarial. O fluxo básico de integração dos SI está representado na Figura 7.



Fonte: Rezende e Abreu (2013, p. 47).

De fato, devido ao aumento expressivo da quantidade de sistemas utilizados pelas organizações nos últimos anos, a integração entre esses sistemas tornou-se imprescindível. Segue algumas das vantagens que fundamentam a escolha pela integração:

- a) automatização e otimização dos processos;
- b) maior controle;
- c) ganho de desempenho e produtividade;
- d) melhora na comunicação;
- e) diminuição dos retrabalhos;

- f) geração de dados mais confiáveis;
- g) redução de custos operacionais.

Entretanto, existe uma infinidade de métodos que podem ser utilizados para integrar dois ou mais sistemas de informação. Sendo assim, esta seção apresentará as principais técnicas que são utilizadas para integrar sistemas.

3.2.1 SQL

A partir de 1990, a integração de sistemas via acesso direto a uma mesma base de dados passou a ser utilizada em larga escala. Sendo assim, devido a simplicidade e facilidade de uso, caso os dados de um SI precisem ser consultados por outro SI, e este tenha permissão de acesso ao banco de dados relacional do outro, a integração via linguagem SQL é altamente recomendada. Além disso, o desempenho desse tipo de integração tende a ser melhor que o dos demais métodos disponíveis (DE SORDI; MARINHO, 2007).

3.2.2 SOA

A arquitetura orientada a serviços, também conhecida como SOA, acrônimo para *Service-Oriented Architecture*, é um modelo arquitetônico de *software* cujas funcionalidades específicas do sistema são disponibilizadas no formato de serviços (ERL, 2009).

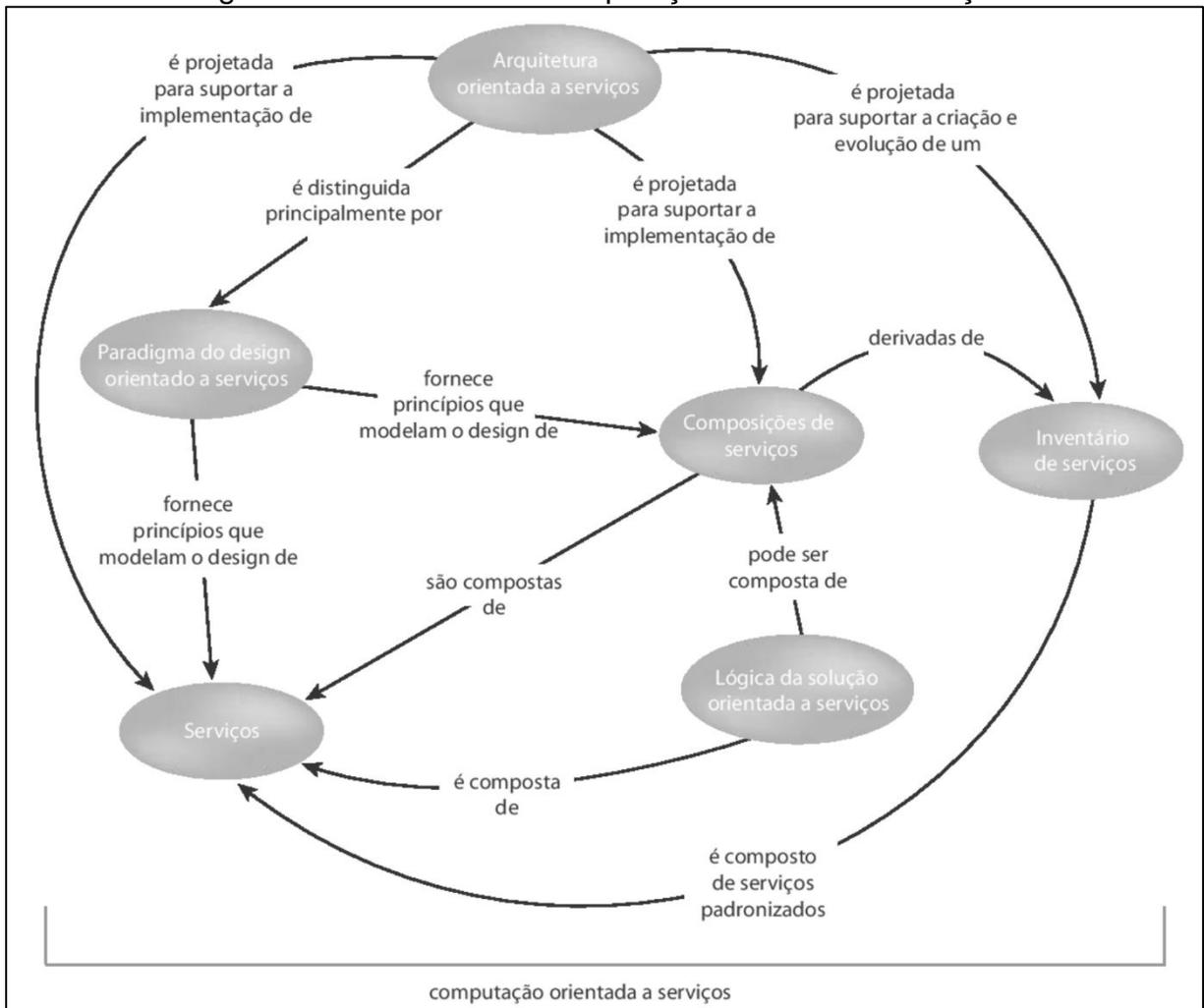
Segundo Erl (2009, p. 24), o objetivo da SOA é “[...] aprimorar a eficiência, a agilidade e a produtividade de uma empresa, posicionando os serviços como os principais meios para que a solução lógica seja representada no suporte à realização dos objetivos estratégicos associados à computação orientada a serviços.”.

É muito importante citar que a SOA é apenas um dos vários elementos da computação orientada a serviços, conforme demonstrado na Figura 8 (ERL, 2009).

De acordo com Erl (2009), os serviços modelados devem ser regidos por contratos padronizados, desacoplados, abstratos, reutilizáveis, autônomos, independentes de estado, visíveis e compostos.

Comumente, essa arquitetura orientada a serviços é implementada utilizando *web services*, porém isso não é obrigatório (ERL, 2009).

Figura 8 – Elementos da computação orientada a serviços



Fonte: Erl (2009, p. 27).

3.2.2.1 API

Uma *Application Programming Interface* (API), ou uma Interface de Programação de Aplicações em português, é uma aplicação que disponibiliza um conjunto de procedimentos estabelecidos e documentados para que usuários ou aplicações possam usufruir das funcionalidades de outra aplicação (SIMONETTI, 2002).

3.2.2.2 REST

O REST (*Representational State Transfer*, ou Transferência Representacional de Estado em português) é um modelo de arquitetura de *software* que define um

conjunto de padrões e princípios a serem seguidos para a implementação de serviços *web* (DE OLIVEIRA *et al.*, 2019; SIMAS *et al.*, 2019).

3.2.2.3 API RESTful

Uma API RESTful é uma API que segue os padrões da arquitetura REST para realizar a troca de dados via *web* (DE OLIVEIRA *et al.*, 2019; SIMAS *et al.*, 2019).

A principal característica de uma API RESTful é que ela se baseia no protocolo HTTP (*HyperText Transfer Protocol*) para realizar a comunicação entre sistemas *web*. Dentro desse contexto, a comunicação entre os sistemas ocorre por meio de requisições em um endereço URL (*Uniform Resource Locator*), também chamado de *endpoint* (DE OLIVEIRA *et al.*, 2019; SIMAS *et al.*, 2019).

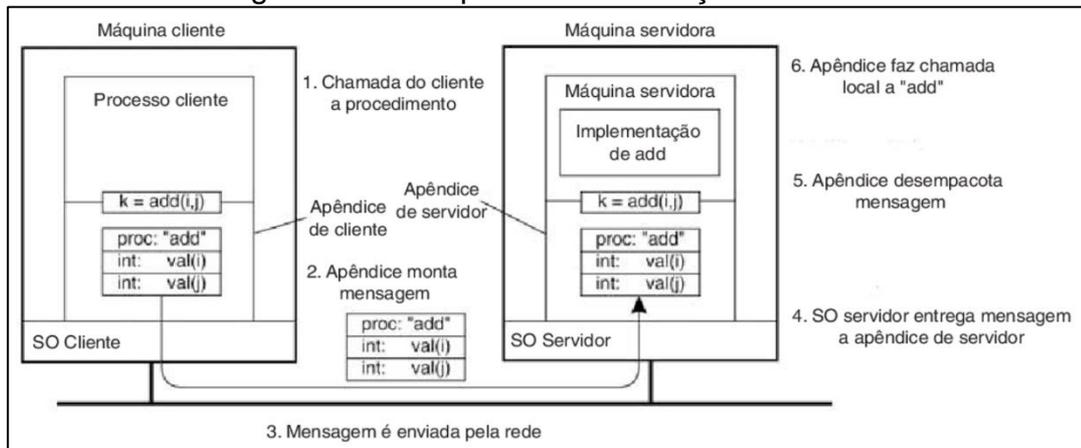
Além disso, o protocolo HTTP fornece uma série de métodos, também referidos como verbos, para que a aplicação possa indicar qual é a finalidade da requisição que está sendo enviada, são eles: GET, POST, PUT, PATCH e DELETE (DE OLIVEIRA *et al.*, 2019; SIMAS *et al.*, 2019).

Outra característica importante do HTTP é que o dado é serializado em um formato de representação padrão. Apesar de permitir a utilização de outros formatos, o formato JSON (*JavaScript Object Notation*) é o mais comum, porque é compacto, simples, legível e rápido (DE OLIVEIRA *et al.*, 2019; SIMAS *et al.*, 2019).

3.2.3 RPC

A chamada de procedimento remoto, ou RPC (acrônimo de *Remote Procedure Call*) como é conhecida, é uma tecnologia de comunicação entre processos, bastante comum em sistemas distribuídos, que permite que programas executem chamadas para procedimentos localizados em outras máquinas. As informações podem ser enviadas do processo chamador para quem foi chamado via parâmetros, e retornadas no resultado do procedimento, conforme exemplificado na Figura 9 (TANENBAUM; STEEN, 2007).

Figura 9 – Exemplo de comunicação via RPC

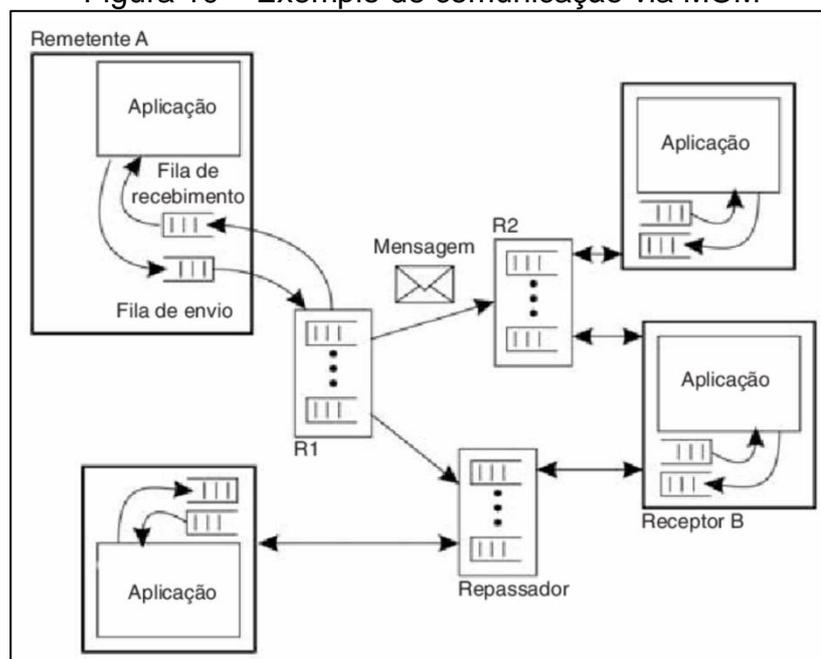


Fonte: Tanenbaum e Steen (2007, p. 78).

3.2.4 MOM

Outro importante método de comunicação entre componentes de *software* que é muito utilizado em sistemas distribuídos é o MOM, ou *Message-Oriented Middleware*, geralmente conhecidos como sistemas de enfileiramento de mensagens. Esse tipo de integração ocorre através da transferência direta de uma mensagem de para um servidor de comunicação, que é responsável por entregar a mensagem ao destinatário de maneira assíncrona, conforme exemplificado na Figura 10 (TANENBAUM; STEEN, 2007).

Figura 10 – Exemplo de comunicação via MOM



Fonte: Tanenbaum e Steen (2007, p. 90).

3.2.5 EDI

Sigla para *Electronic Data Interchange*, ou intercâmbio eletrônico de dados em português, é um método de troca de informações entre dois sistemas via arquivos de texto padronizados. Aqui, vale a pena ressaltar, que os arquivos podem ser trocados entre duas organizações diferentes (MARTINS, 2005).

3.3 CONSIDERAÇÕES FINAIS

Após o estudo realizado sobre os principais sistemas de informação e técnicas de integração existentes, pôde-se concluir que o método de integração via API's RESTful é o mais recomendado para assegurar os dados trocados entre as aplicações, porém nem sempre será possível utilizá-lo.

4 ESPECIFICAÇÃO

Este capítulo especifica o *software* proposto como solução para o problema de pesquisa abordado na seção 1.1 deste trabalho, limitando-se em descrever as etapas da fase I deste projeto.

De um modo geral, esse problema específico foi resolvido através do desenvolvimento de um *chatbot* que é capaz de responder as perguntas mais frequentes sobre os pedidos de venda da Sazi.

Ademais, para responder ao usuário de uma forma rápida e precisa, o *chatbot* desenvolvido teve que ser integrado ao sistema *Drummer APS*, que já era utilizado pela organização em questão.

Também foram respeitados todos os princípios de governança do setor de TI da empresa, bem como, a preferência por produtos, serviços e plataformas de desenvolvimento da *Microsoft*. Inclusive, o *chatbot* foi disponibilizado na ferramenta *Microsoft Teams*, pois esse é o canal oficial de comunicação interna do Grupo Sazi.

A presente proposta de solução especifica os requisitos do sistema e o projeto do *software* que foi desenvolvido. A especificação dos requisitos conta com a escolha dos usuários chave e a elicitación dos requisitos funcionais e não funcionais do sistema. O projeto do *software* apresenta o comportamento esperado da aplicação, os componentes de *software* de terceiros integrados, a arquitetura do sistema proposto e o modelo de linguagem definido.

4.1 DEFINIÇÃO DE USUÁRIOS CHAVE

Concluída a etapa de estudos e fundamentação teórica para embasamento do presente trabalho, o primeiro passo para o sucesso desse projeto foi a definição dos usuários chave, cujo papel principal foi o de fornecer as informações referentes às áreas de negócio envolvidas.

Dessa forma, no dia 13/05/2020, foi oficializado o convite para dois funcionários do Grupo Sazi que são referência nas suas áreas de atuação:

- a) **Angélica de Carvalho dos Santos**: programadora de produção sênior e líder do setor de PCP;
- b) **Marcio Silveira**: técnico comercial sênior e coordenador da área comercial.

Ambos os colaboradores aceitaram imediatamente o convite para participar desse projeto. Sendo assim, no mesmo instante, já foram agendadas entrevistas para elicitação dos requisitos funcionais e não funcionais do sistema.

4.2 ELICITAÇÃO DE REQUISITOS

Os requisitos funcionais e não funcionais do *chatbot* foram elicitados por meio da técnica de entrevista individual aberta com os usuários chave definidos na etapa anterior (SOMMERVILLE, 2018). Para tal, foram organizadas e realizadas quatro reuniões por videoconferência com o auxílio da ferramenta *Microsoft Teams*.

As duas primeiras reuniões tiveram a finalidade de entrevistar os usuários chave para compreender melhor o cenário atual do problema de pesquisa e elicitar todos os requisitos funcionais e não funcionais do projeto.

Já os dois últimos encontros tiveram o objetivo de validar as perguntas mais frequentes sobre pedidos de venda que haviam sido levantadas nas entrevistas anteriores, bem como, alinhar a expectativa dos usuários chave quanto ao produto final que será gerado por este trabalho.

Além disso, foi utilizada uma adaptação da matriz SIPOC (*Supplier, Input, Process, Output, Customer*) como ferramenta para guiar as entrevistas realizadas com os usuários chave e mapear as perguntas mais frequentes sobre pedidos de venda que são respondidas pelos setores, conforme especificadas no Quadro 2. No entanto, tradicionalmente, a matriz SIPOC é uma ferramenta simples que é aplicada para o mapeamento de processos de negócios (DA SILVA, 2015).

Segue uma breve explicação de como foi preenchida cada coluna da matriz SIPOC nas entrevistas:

- a) **Supplier (Fornecedor)**: quem responde à pergunta;
- b) **Input (Entrada)**: quais informações são necessárias para que o fornecedor consiga responder à pergunta;
- c) **Process (Processo)**: qual é a pergunta;
- d) **Output (Saída)**: quais informações são esperadas pelo cliente na resposta;
- e) **Customer (Cliente)**: quem pergunta e aguarda por uma resposta.

Quadro 2 – Matriz SIPOC com as perguntas mais frequentes sobre pedidos

Supplier	Input	Process	Output	Customer
PCP	Item e Data	É possível antecipar a entrega do item # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Item e Data	É possível antecipar a entrega de # peças do item # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Pedido e Data	É possível antecipar a entrega do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Pedido e Data	É possível antecipar a entrega de # peças do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Item, Pedido e Data	É possível antecipar a entrega do item # do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Item, Pedido e Data	É possível antecipar a entrega de # peças do item # do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Item	Os pedidos do item # serão entregues no prazo?	Sim/Não com Atraso	Comercial
PCP	Pedido	O pedido # será entregue no prazo?	Sim/Não com Atraso	Comercial
PCP	Item e Pedido	O item # do pedido # será entregue no prazo?	Sim/Não com Atraso	Comercial
PCP	Cliente	Quais pedidos do cliente # irão atrasar?	Pedidos com Atraso	Comercial
PCP		Quais pedidos irão atrasar?	Pedidos com Atraso	Comercial
PCP	Item	Os pedidos do item # já estão em estoque?	Sim/Não com Estoque	Comercial
PCP	Pedido	O pedido # já está em estoque?	Sim/Não com Estoque	Comercial
PCP	Item e Pedido	O item # do pedido # já está em estoque?	Sim/Não com Estoque	Comercial
PCP	Cliente	Quais pedidos do cliente # já estão em estoque?	Pedidos em Estoque	Comercial
PCP		Quais pedidos já estão prontos no estoque?	Pedidos em Estoque	Comercial
Comercial	Pedido	Qual é a data de entrega original do pedido #?	Data de Entrega Original	PCP
PCP	Pedido	Qual é data de entrega reprogramada do pedido #?	Data de Entrega Reprogramada	Comercial
Comercial	Item	Quais são os pedidos pendentes do item #?	Pedidos Pendentes	PCP
Comercial	Cliente	Quais são os pedidos pendentes do cliente #?	Pedidos Pendentes	Comercial

Fonte: Autor (2020).

Os requisitos funcionais e não funcionais elicitados, e validados com os usuários chave, encontram-se listados nos Quadros 3 e 4, respectivamente.

Quadro 3 – Requisitos funcionais do *chatbot*

Identificador	Descrição
RF01	Responder as perguntas mais frequentes sobre pedidos de venda, conforme detalhado no Quadro 2.
RF02	Consultar as informações de um pedido no sistema <i>Drummer APS</i> .
RF03	Alterar a data de entrega de um pedido no sistema <i>Drummer APS</i> .
RF04	Alterar a quantidade de um pedido no sistema <i>Drummer APS</i> .
RF05	Simular a antecipação de um pedido no sistema <i>Drummer APS</i> .

Fonte: Autor (2020).

Quadro 4 – Requisitos não funcionais do *chatbot*

Identificador	Descrição
RNF01	Integrar com o sistema <i>Drummer APS</i> .
RNF02	Apresentar bom desempenho.
RNF03	Ser intuitivo para o usuário.
RNF04	Garantir a manutenibilidade do sistema.
RNF05	Garantir a segurança dos dados.
RNF06	Criar um usuário para o <i>chatbot</i> no sistema <i>Drummer APS</i> .
RNF07	Criar um ambiente para o <i>chatbot</i> no sistema <i>Drummer APS</i> .
RNF08	Respeitar os princípios de governança de TI da empresa.
RNF09	Priorizar o uso de soluções oferecidas pela <i>Microsoft</i> .
RNF10	Publicar o <i>chatbot</i> no canal <i>Microsoft Teams</i> .

Fonte: Autor (2020).

4.3 PROJETO DE SOFTWARE

Nessa etapa, será projetado um *software* para atender todos os requisitos funcionais e não funcionais elicitados na etapa antecessora, utilizando a UML na construção de todos os artefatos que são necessários para dar início ao processo de desenvolvimento da aplicação. Conforme Larman (2007, p. 32, grifo do autor), “A **UML** é uma notação padrão de diagramação.”.

Visando facilitar a compreensão do *software* projetado, optou-se por utilizar dois diagramas da UML:

- a) **diagrama de atividades**: diagrama que descreve o comportamento do sistema;
- b) **diagrama de implantação**: diagrama que descreve a estrutura do sistema.

Além da modelagem comportamental e estrutural do sistema, também serão especificados os componentes de *software* de terceiros que compõem esta solução e o modelo de linguagem que será utilizado na fase seguinte.

4.3.1 Modelagem comportamental

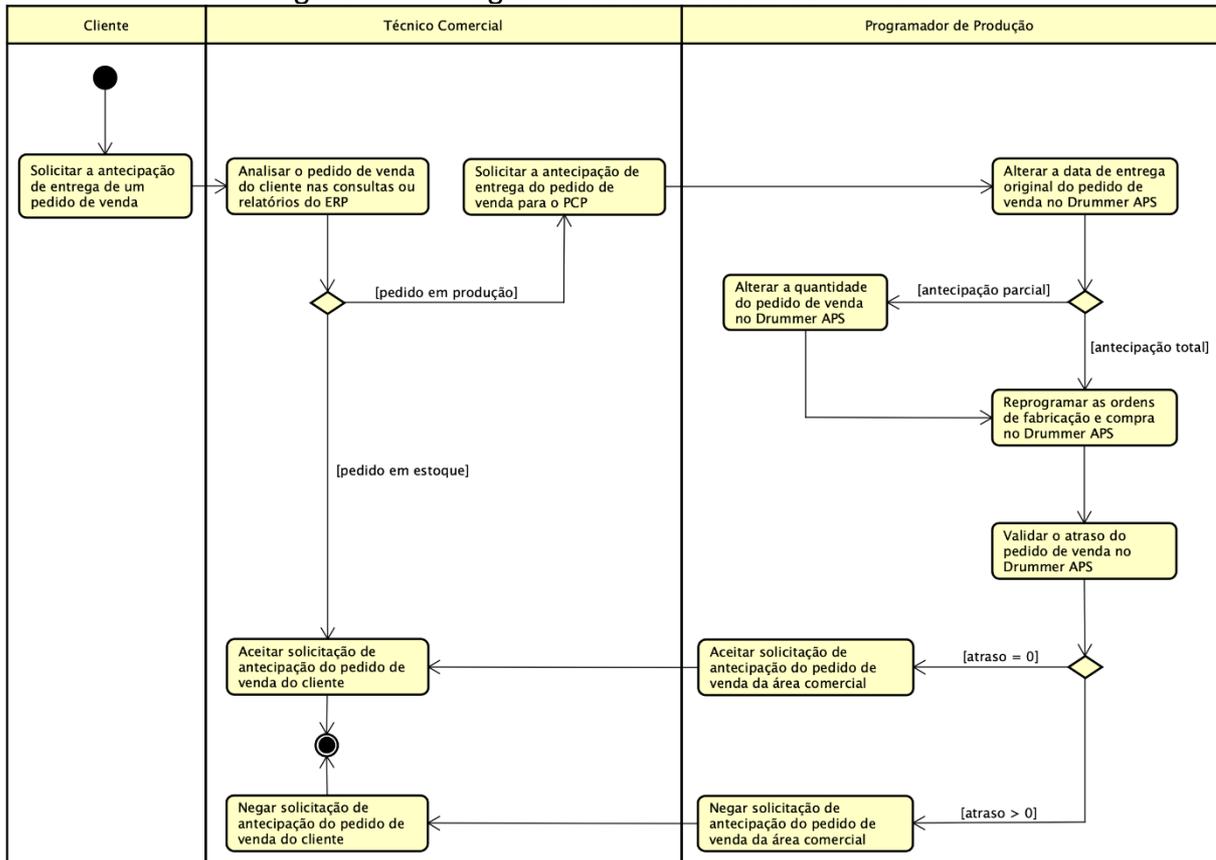
Para exemplificar os aspectos comportamentais do sistema que será desenvolvido neste trabalho, foram construídos dois diagramas de atividades, levando em consideração a pergunta mais frequente sobre pedidos de venda que é respondida atualmente pela empresa analisada.

O primeiro diagrama de atividades apresentado na Figura 11 demonstra o fluxo atual do processo que é executado quando um cliente pergunta para um técnico comercial da empresa se é possível antecipar a entrega de um determinado pedido de venda, segundo dados levantados com o usuário chave da área comercial, Sr. Marcio Silveira.

Ao receber essa pergunta, normalmente por *e-mail*, o técnico comercial analisa o pedido de venda do cliente nas consultas ou relatórios do sistema ERP para entender qual é a situação do pedido. Caso haja quantidade disponível em estoque dos itens para atender ao pedido de venda, o técnico comercial responde para o cliente que o pedido já está pronto e que é possível antecipar a entrega. Entretanto, caso os itens do pedido ainda estejam em processo de fabricação ou compra, o

técnico comercial encaminha a solicitação do cliente para um programador do setor de PCP realizar uma análise mais profunda considerando a capacidade finita dos recursos da fábrica.

Figura 11 – Diagrama de atividades sem *chatbot*



Fonte: Autor (2020).

Ainda, de acordo com a usuária chave do setor de PCP da empresa, Sra. Angélica de Carvalho dos Santos, o método recomendado para realizar essa análise é executando o seguinte procedimento no sistema *Drummer APS*:

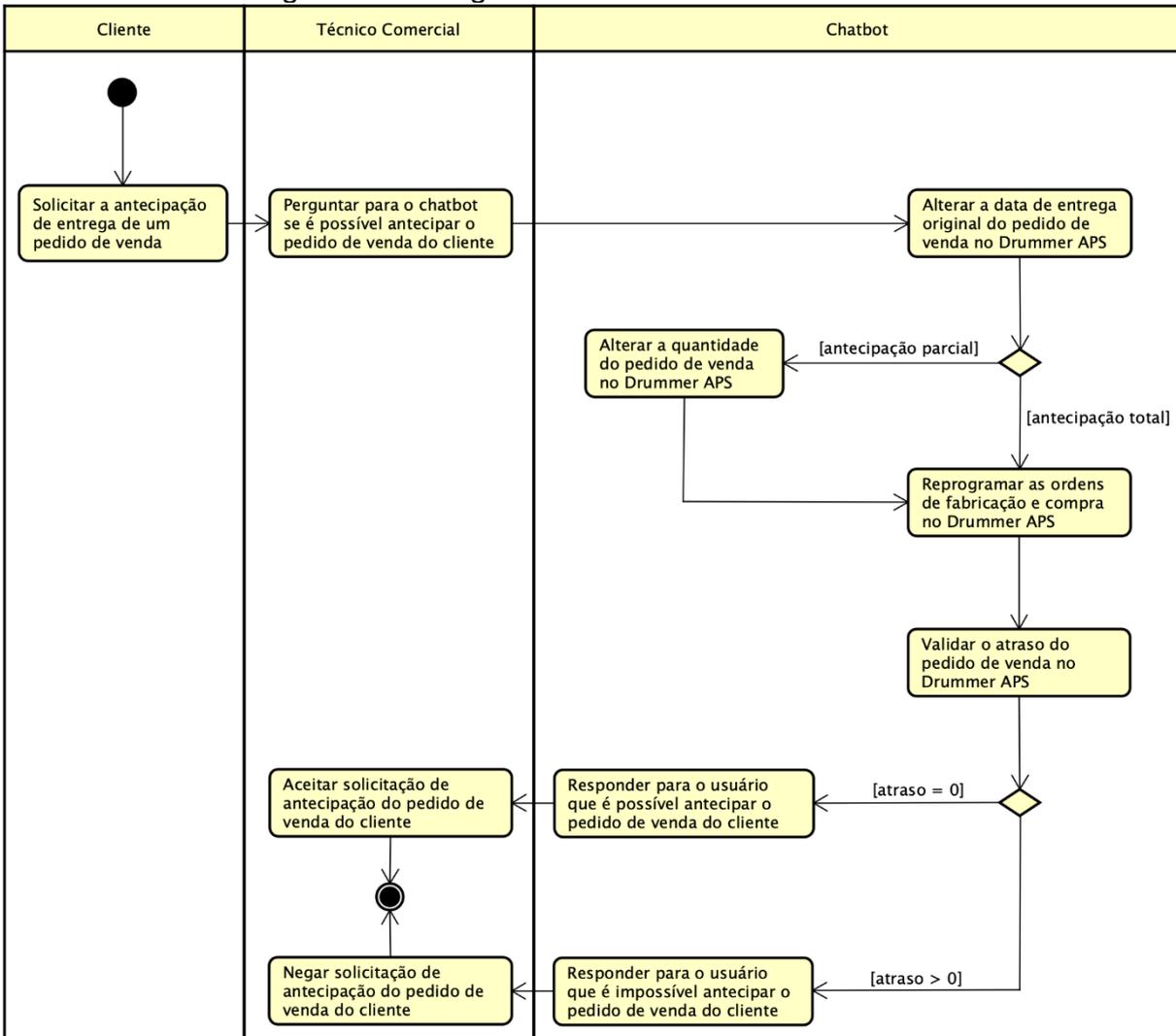
- alterar a data de entrega original do pedido de venda para a nova data solicitada pelo cliente;
- alterar a quantidade do pedido de venda se for uma solicitação de antecipação parcial, ou seja, se o cliente deseja receber apenas parte das peças do pedido antes do prazo inicialmente acordado;
- reprogramar as ordens de fabricação e compra;
- validar o atraso calculado pelo sistema para o novo pedido de venda.

Nesse caso, se o sistema *Drummer APS* indicar atraso na entrega do pedido de venda analisado, então o programador de produção não pode aceitar a solicitação

de antecipação. Caso contrário, apesar de haver outros riscos diretamente relacionados com esse processo que não serão discutidos neste trabalho, a solicitação de antecipação do pedido de venda pode ser aceita sem a incidência de grandes prejuízos para a fábrica.

Nesse ponto é importante citar que, comumente, essa interação entre o técnico comercial e o programador de produção também é realizada via mensagem eletrônica. Ou seja, após a análise da solicitação comercial, o programador de produção responde o *e-mail* apenas para o técnico comercial, e este, quando ver a resposta do colega, finalmente responde para o cliente. Resumindo, esse processo pode ser bastante demorado, causando insatisfação em todas as partes interessadas.

Figura 12 – Diagrama de atividades com *chatbot*



Fonte: Autor (2020).

Em contrapartida, o diagrama de atividades apresentado na Figura 12 demonstra como será o mesmo processo após a implantação do *chatbot*. Além disso, nesse mesmo diagrama, é possível ter uma ideia de como será a interação dos usuários com o sistema proposto por este trabalho.

Conforme pode ser notado na Figura 12, nesse novo cenário, o *chatbot* ficará responsável por todas as atividades que antes eram executadas por um programador de produção, e o técnico comercial não precisará mais consultar o pedido de venda no sistema ERP da empresa.

Desse modo, acredita-se que a implantação do *chatbot* reduzirá significativamente o tempo de resposta ao cliente final, dispensará o envolvimento de um programador de produção no processo, liberando mão de obra especializada para a execução de outras atividades que a empresa julgar mais importantes, e aumentará a satisfação de todas as partes envolvidas no processo.

4.3.2 Componentes de software de terceiros

Esta seção introduz os principais componentes de *software* de terceiros que foram acoplados na solução proposta, antes do modelo estrutural ser discutido.

4.3.2.1 Drummer APS

O Grupo Sazi investe em inúmeros sistemas especialistas acoplados ao ERP para atender a processos de negócio específicos e customizados, sempre mantendo o foco na melhoria contínua.

Dessa forma, o primeiro componente de terceiro reutilizado por este trabalho é o sistema de informação *Drummer APS*, da empresa Linter Sistemas de São Paulo, que é o sistema APS adotado pelo Grupo Sazi como solução para todos os processos de negócio da área de PPCPM.

Segundo a Linter (2020), o *software Drummer APS* possibilita uma rápida reação da empresa perante as constantes mudanças que ocorrem no processo produtivo e na demanda do mercado, configurando-se como uma poderosa ferramenta nas tomadas de decisões.

Um dos principais diferenciais do sistema *Drummer APS* é que o mesmo permite a criação de múltiplos cenários e a rápida execução de inúmeras simulações

do processo produtivo, o que torna o *software* ideal para responder as perguntas dos usuários relacionadas ao reposicionamento da carteira de clientes (LINTER, 2020).

Além disso, algumas das informações que são essenciais para responder aos questionamentos dos usuários relacionados à antecipação ou postergação de entrega de um determinado pedido de venda são cadastradas exclusivamente no *Drummer* APS pelos usuários do PCP, como por exemplo, a capacidade dos recursos produtivos.

Ou seja, a integração desse sistema APS, que já é utilizado pela organização, com o *chatbot* desenvolvido por este trabalho é fundamental para que seja possível responder as perguntas dos usuários referentes aos pedidos de venda da Sazi. Caso contrário, as respostas serão inadequadas, incompletas e insatisfatórias.

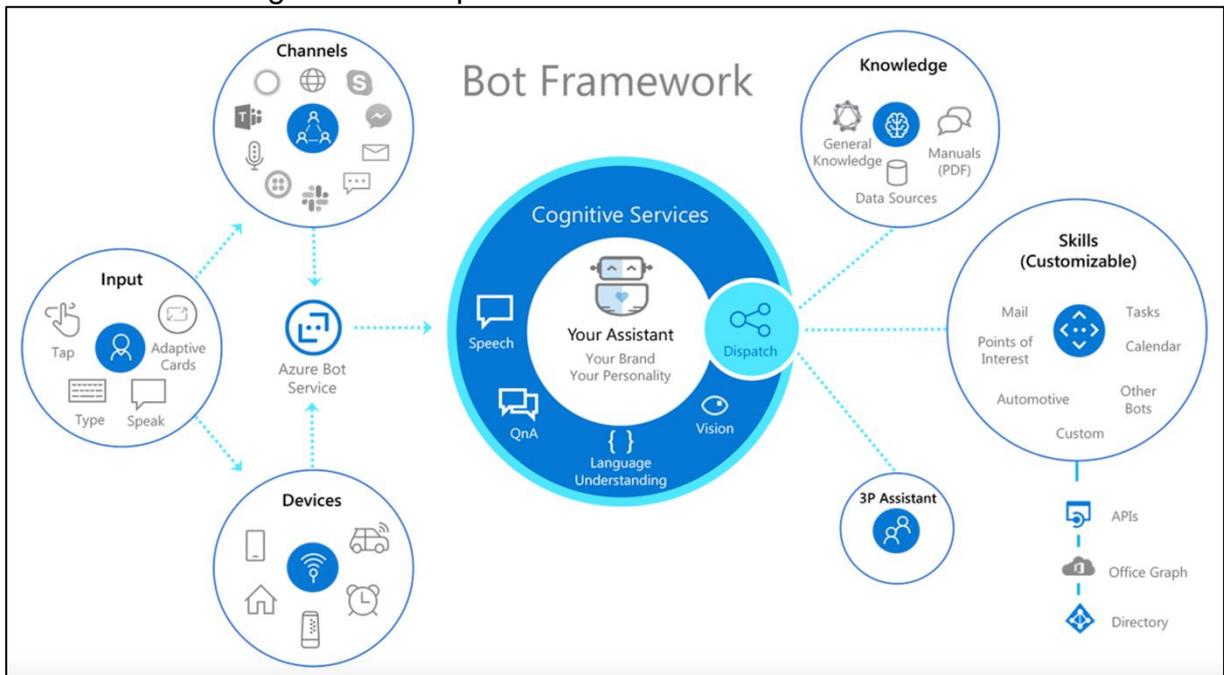
4.3.2.2 Microsoft Bot Framework

Visando atender aos requisitos não funcionais RNF08 e RNF09, buscou-se soluções para o desenvolvimento de *chatbots* específicas da *Microsoft*, porque a empresa já é parceira desse fornecedor há muitos anos, e a equipe de TI possui maior conhecimento nessa plataforma de produtos e serviços, garantindo assim, a manutenibilidade da solução entregue por este projeto.

Outro requisito não funcional essencial para a tomada dessa decisão foi o RNF10, cuja especificação determina que o *chatbot* desenvolvido deve ser publicado no canal *Microsoft Teams*, que também é da *Microsoft*, visto que todas as áreas já utilizam essa plataforma corporativa unificada de comunicação e colaboração interna.

Após pesquisa dos produtos e serviços oferecidos pela *Microsoft* que pudessem auxiliar no desenvolvimento deste trabalho, o *Software Development Kit* (SDK), ou kit de desenvolvimento de *software* em português, *Microsoft Bot Framework* mostrou-se ser o mais adequado nesse caso.

O conjunto de ferramentas do *Microsoft Bot Framework* fornece um ambiente integrado repleto de recursos exclusivos para a construção de *chatbots* empresariais, conforme ilustrado na Figura 13 (MICROSOFT, 2020c).

Figura 13 – Arquitetura do *Microsoft Bot Framework*

Fonte: Microsoft (2020c).

4.3.2.2.1 Serviços de PLN

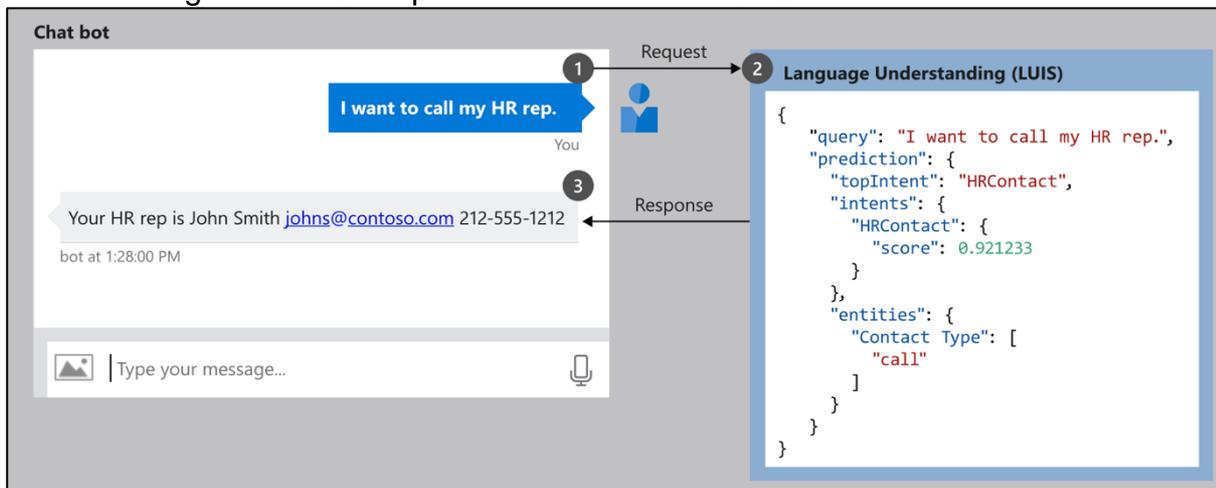
O *Microsoft Bot Framework* oferece diversos serviços cognitivos com IA para o desenvolvedor de *software*, dentre eles, dois serviços *web* são específicos para o processamento de linguagem natural (PLN): *Language Understanding Intelligent Service* (LUIS) e *QnA Maker*. Porém, cada um possui uma finalidade diferente, de acordo com a própria *Microsoft* (2020c).

O LUIS é um serviço *web* de PLN que permite que o desenvolvedor crie seus próprios modelos de linguagem personalizados para adicionar inteligência artificial ao seu aplicativo. Os modelos de linguagem aprendidos pela máquina capturam o texto de entrada não estruturado do usuário e retornam uma resposta formatada em JSON, que contém o enunciado, as possíveis intenções classificadas e as entidades extraídas da expressão (MICROSOFT, 2020b).

A Figura 14 exemplifica o funcionamento do LUIS em um *chatbot*. No exemplo, a aplicação cliente (*chatbot*) envia o enunciado do usuário para o *endpoint* do LUIS como uma requisição HTTP. O LUIS recebe essa requisição, processa e retorna uma resposta no formato JSON para a aplicação cliente, contendo o enunciado, a intenção e a entidade do texto digitado pelo usuário. O *chatbot*, por sua

vez, usa a resposta JSON para tomar decisões sobre como atender às solicitações do usuário. Essas decisões podem incluir até mesmo chamadas para outros serviços.

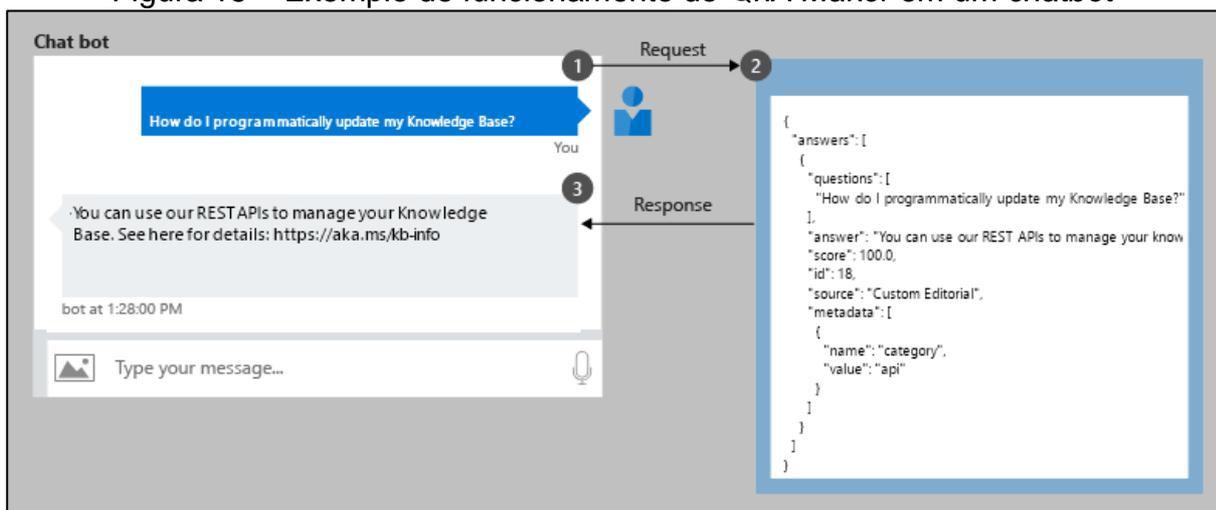
Figura 14 – Exemplo de funcionamento do LUIS em um *chatbot*



Fonte: Microsoft (2020b).

Por outro lado, o *QnA Maker* é um serviço *web* de PLN baseado em nuvem que cria uma camada de perguntas e respostas personalizadas sobre os dados, denominada base de conhecimento. Isso permite que o *chatbot* envie uma pergunta ao *QnA Maker* e receba uma resposta sem a necessidade de analisar e interpretar a intenção da pergunta (MICROSOFT, 2020e).

Figura 15 – Exemplo de funcionamento do *QnA Maker* em um *chatbot*



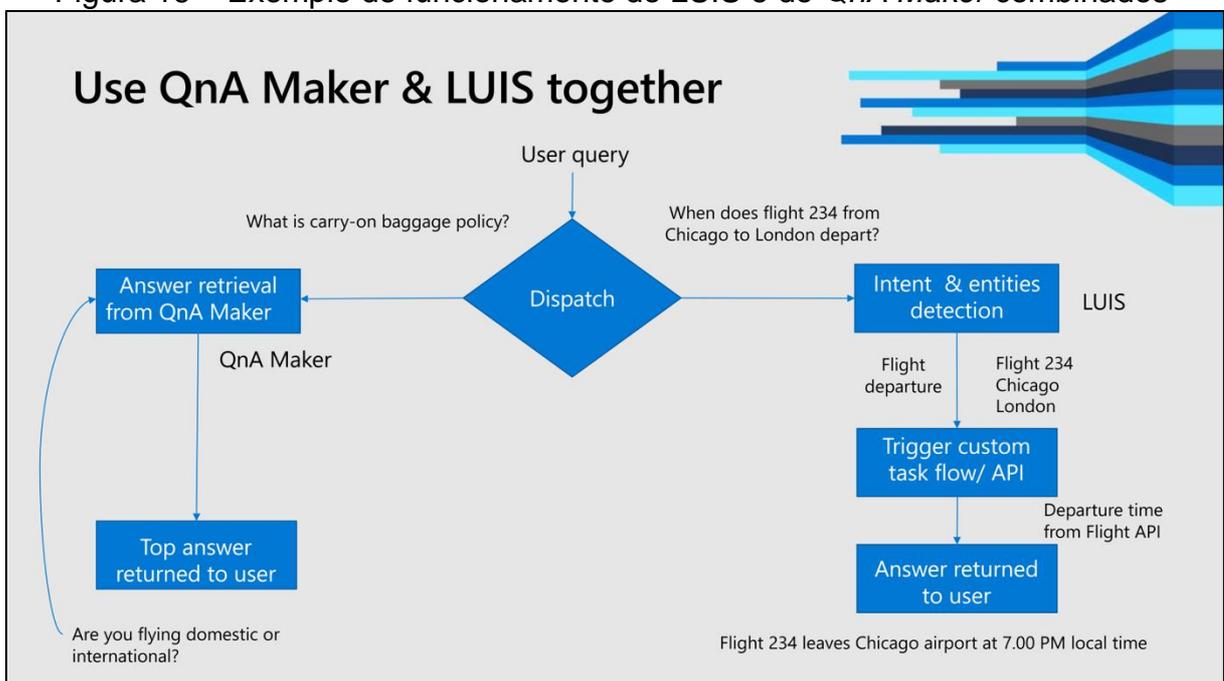
Fonte: Microsoft (2020e).

A Figura 15 exemplifica o funcionamento do *QnA Maker* em um *chatbot*. No exemplo, a aplicação cliente (*chatbot*) envia a pergunta do usuário para o *endpoint* do

QnA Maker como uma requisição HTTP. O *QnA Maker* recebe essa requisição, processa a linguagem natural do usuário, pesquisa por uma resposta na base de conhecimento e retorna a melhor resposta encontrada no formato JSON para a aplicação cliente. O *chatbot*, por sua vez, usa a resposta JSON para tomar decisões sobre como continuar a conversa. Essas decisões podem incluir até mesmo mostrar a resposta principal e apresentar mais opções para que o usuário possa refinar a sua pesquisa.

Outra opção sugerida pela *Microsoft* (2020c) é a utilização desses dois serviços combinados, quando a base de dados de conhecimento estiver incompleta ou o sistema necessitar de informações complementares. Assim, por exemplo, quando o usuário faz uma pergunta para o *chatbot*, a primeira ação da aplicação pode ser consultar a base de dados de conhecimento do *QnA Maker*, caso não encontre a resposta, ele realiza uma nova consulta com o LUIS. A Figura 16 exemplifica o funcionamento do uso combinado desses dois serviços de PLN.

Figura 16 – Exemplo de funcionamento do LUIS e do *QnA Maker* combinados



Fonte: Microsoft (2020b).

Segundo a *Microsoft* (2020c), enquanto o LUIS é recomendado para qualquer tipo de conversa com o usuário em que se deseja extrair as intenções e as entidades presentes nos enunciados, o *QnA Maker* é indicado apenas para *chatbots* cuja resposta para uma pergunta do usuário possa ser localizada em uma base de dados

de conhecimento estática, personalizada e criada no formato de FAQ (*Frequently Asked Questions*).

Desse modo, como o *chatbot* desenvolvido por este trabalho precisa ser bem dinâmico, optou-se por utilizar apenas o serviço *web* LUIS para processar a linguagem natural do usuário.

4.3.2.2 Canais

Outra característica muito interessante do *Microsoft Bot Framework* é a facilidade em conectar o *chatbot* desenvolvido aos principais meios de comunicação utilizados pelas empresas. Essa conexão entre o *chatbot* e os aplicativos de comunicação é chamada de canal (MICROSOFT, 2020c).

Conforme sinalizado pela *Microsoft* (2020c), o *chatbot* pode ser publicado em múltiplos canais através do *Azure Bot Service*, permitindo que o desenvolvedor não precise se preocupar com o canal, visto que o serviço fica responsável por normalizar e controlar o fluxo das mensagens trocadas com a ferramenta de comunicação escolhida.

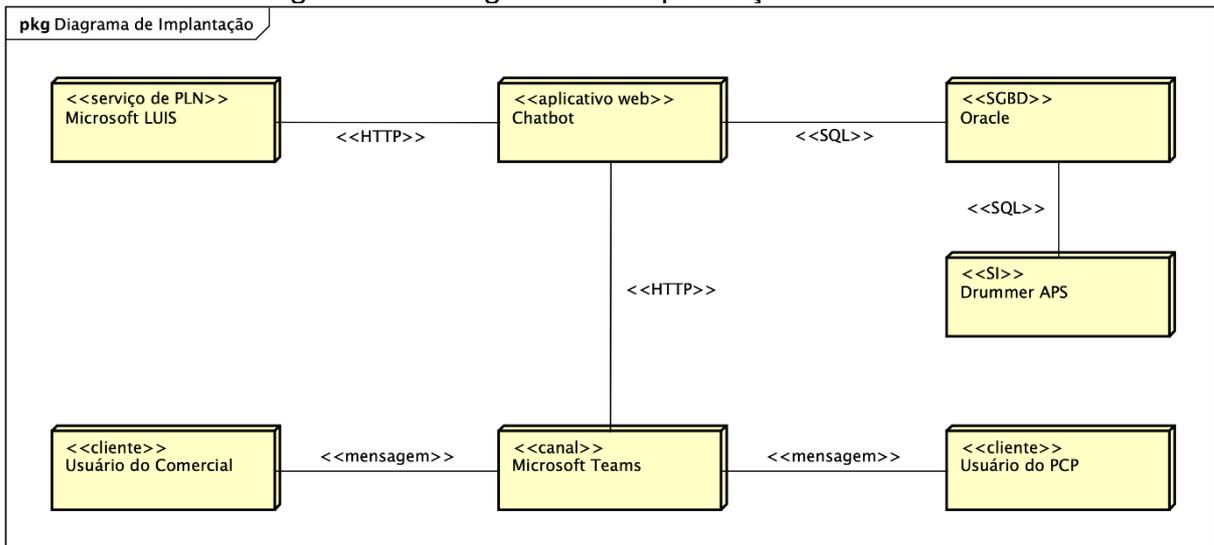
Dentre os diversos canais integrados com o *Azure Bot Service*, está o *Microsoft Teams*, que é uma plataforma unificada de comunicação e colaboração para agilizar o trabalho em equipe (MICROSOFT, 2020d).

4.3.3 Modelagem estrutural

Considerando todos os requisitos funcionais e não funcionais deste projeto, foi modelado um *chatbot* para solucionar o problema de pesquisa discutido na introdução deste trabalho.

O diagrama de implantação da Figura 17 apresenta uma visão geral da arquitetura física e lógica do *chatbot* proposto. Nesse mesmo diagrama também é possível analisar como ocorrerá a comunicação do aplicativo com os variados componentes de *software* de terceiros acoplados ao projeto (LARMAN, 2007).

Como pode ser observado na Figura 17, o *chatbot* desenvolvido neste trabalho é um aplicativo *web* que funciona como uma interface de comunicação entre os usuários e o sistema APS utilizado pelo Grupo Sazi.

Figura 17 – Diagrama de implantação do *chatbot*

Fonte: Autor (2020).

A integração do *chatbot* com um dos principais sistemas de informação adotados pela organização em questão, o *Drummer APS*, será realizada via acesso direto ao banco de dados *Oracle*. Essa técnica de integração foi escolhida, porque todos os demais sistemas da empresa já são integrados dessa maneira. Inclusive, o próprio sistema *Drummer APS* importa os dados do ERP usando o mesmo método. Contudo, para garantir a segurança da informação, o *chatbot* se conectará ao SGBD *Oracle* com um usuário específico, cujas permissões de acesso serão totalmente controladas pelo departamento de TI da empresa.

Por fim, o *chatbot* desenvolvido será publicado no canal *Microsoft Teams*, porque essa já é a ferramenta de comunicação interna adotada pelo Grupo Sazi.

Assim, tanto os usuários da área comercial, quanto os usuários do setor de PCP, poderão conversar, simultaneamente, com o *chatbot* desenvolvido por meio da troca de mensagens de texto em um *chat* do *Microsoft Teams*, que se comunicará com o aplicativo *web* através de requisições e respostas no padrão HTTP.

4.3.4 Modelagem linguística

A última atividade da fase I deste projeto foi especificar um modelo de linguagem para o *chatbot*. Para isso, foi necessário identificar e extrair os enunciados, as intenções e as entidades das perguntas mais frequentes sobre pedidos de venda que foram apuradas na etapa anterior de elicitação de requisitos com os usuários

chave e apresentadas no Quadro 2. Aliás, a utilização dessa técnica nas entrevistas de levantamento dos requisitos facilitou muito esse processo.

Primeiramente, os enunciados foram extraídos da coluna “*Process*” do Quadro 2, tal como haviam sido redigidos.

Após, os enunciados foram agrupados pela coluna “*Output*”, que indica qual é a resposta esperada pelo usuário para a pergunta. O Quadro 5 demonstra esse agrupamento.

Quadro 5 – Matriz SIPOC agrupada pela saída esperada

Supplier	Input	Process	Output	Customer
PCP	Item e Data	É possível antecipar a entrega do item # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Item e Data	É possível antecipar a entrega de # peças do item # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Pedido e Data	É possível antecipar a entrega do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Pedido e Data	É possível antecipar a entrega de # peças do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Item, Pedido e Data	É possível antecipar a entrega do item # do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Quantidade, Item, Pedido e Data	É possível antecipar a entrega de # peças do item # do pedido # para o dia DD/MM/AAAA?	Sim/Não com Motivo	Comercial
PCP	Item	Os pedidos do item # serão entregues no prazo?	Sim/Não com Atraso	Comercial
PCP	Pedido	O pedido # será entregue no prazo?	Sim/Não com Atraso	Comercial
PCP	Item e Pedido	O item # do pedido # será entregue no prazo?	Sim/Não com Atraso	Comercial
PCP	Cliente	Quais pedidos do cliente # irão atrasar?	Pedidos com Atraso	Comercial
PCP		Quais pedidos irão atrasar?	Pedidos com Atraso	Comercial
PCP	Item	Os pedidos do item # já estão em estoque?	Sim/Não com Estoque	Comercial
PCP	Pedido	O pedido # já está em estoque?	Sim/Não com Estoque	Comercial
PCP	Item e Pedido	O item # do pedido # já está em estoque?	Sim/Não com Estoque	Comercial
PCP	Cliente	Quais pedidos do cliente # já estão em estoque?	Pedidos em Estoque	Comercial
PCP		Quais pedidos já estão prontos no estoque?	Pedidos em Estoque	Comercial
Comercial	Pedido	Qual é a data de entrega original do pedido #?	Data de Entrega Original	PCP
PCP	Pedido	Qual é data de entrega reprogramada do pedido #?	Data de Entrega Reprogramada	Comercial
Comercial	Item	Quais são os pedidos pendentes do item #?	Pedidos Pendentes	PCP
Comercial	Cliente	Quais são os pedidos pendentes do cliente #?	Pedidos Pendentes	Comercial

Fonte: Autor (2020).

Assim, percebeu-se que existiam apenas quatro intenções que o *chatbot* precisaria compreender nas perguntas mais frequentes sobre pedidos de venda, conforme detalhado no Quadro 6.

Quadro 6 – Intenções e enunciados extraídos da matriz SIPOC

(continua)

Intenções	Enunciados
Consultar Antecipação	É possível antecipar a entrega do item # para o dia DD/MM/AAAA?
	É possível antecipar a entrega de # peças do item # para o dia DD/MM/AAAA?
	É possível antecipar a entrega do pedido # para o dia DD/MM/AAAA?
	É possível antecipar a entrega de # peças do pedido # para o dia DD/MM/AAAA?
	É possível antecipar a entrega do item # do pedido # para o dia DD/MM/AAAA?
	É possível antecipar a entrega de # peças do item # do pedido # para o dia DD/MM/AAAA?
Consultar Atraso	Os pedidos do item # serão entregues no prazo?
	O pedido # será entregue no prazo?

(conclusão)

Intenções	Enunciados
Consultar Atraso	O item # do pedido # será entregue no prazo?
	Quais pedidos do cliente # irão atrasar?
	Quais pedidos irão atrasar?
Consultar Estoque	Os pedidos do item # já estão em estoque?
	O pedido # já está em estoque?
	O item # do pedido # já está em estoque?
	Quais pedidos do cliente # já estão em estoque?
	Quais pedidos já estão prontos no estoque?
Consultar Pedido	Qual é a data de entrega original do pedido #?
	Qual é data de entrega reprogramada do pedido #?
	Quais são os pedidos pendentes do item #?
	Quais são os pedidos pendentes do cliente #?

Fonte: Autor (2020).

Além dessas quatro intenções principais, também se identificou a necessidade do *chatbot* interagir com outras duas intenções para socializar com o usuário e traçar sua própria personalidade, que são: cumprimentar e despedir.

Por fim, mas não menos importante, as entidades foram especificadas com base na segunda coluna (“*Input*”) do Quadro 2, porque nessa coluna foram registrados os dados complementares que o cliente (usuário) precisa fornecer para que o fornecedor (*chatbot*) consiga responder à pergunta assertivamente, concorde relacionado a seguir:

- a) nome do cliente;
- b) código do cliente;
- c) código do item;
- d) número do pedido;
- e) linha do pedido;
- f) quantidade;
- g) data.

5 IMPLEMENTAÇÃO

Finalizada a fase I do presente trabalho, deu-se início a fase II do projeto com a implementação do *chatbot* que havia sido apenas idealizado até então. Este capítulo relata como foi desenvolvido, testado e avaliado o *chatbot* especificado no Capítulo 4 e proposto para solucionar o problema informado na seção 1.1.

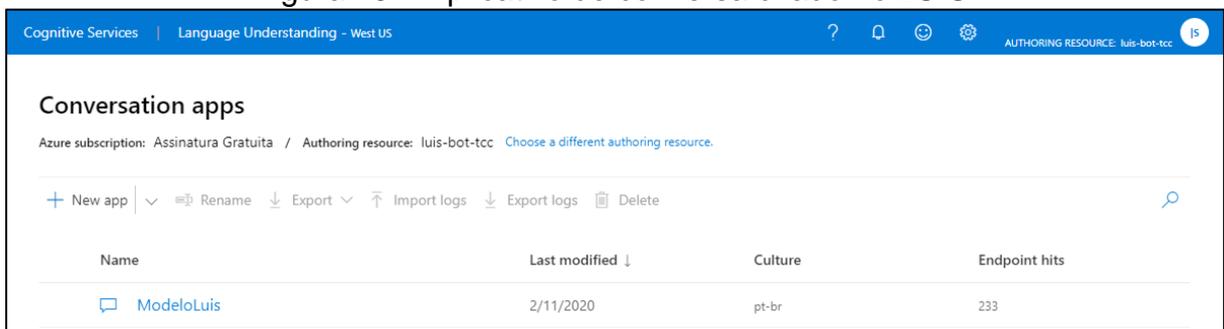
5.1 DESENVOLVIMENTO

O *chatbot* é um aplicativo *web* que foi implementado utilizando a linguagem de programação orientada a objetos *C Sharp (C#)*, a plataforma *ASP.NET* da *Microsoft*, o *SDK Microsoft Bot Framework* e os serviços de nuvem do *Microsoft Azure*, respeitando o requisito RNF09. Nesta seção será descrita a etapa de desenvolvimento do *chatbot*.

5.1.1 Criação de aplicativo no LUIS

De posse das intenções e entidades especificadas na seção 4.3.4 deste texto, a primeira atividade da fase II deste projeto foi a criação de um aplicativo de conversa no portal do LUIS (MICROSOFT, 2020b), contendo um modelo de linguagem natural de domínio específico para que o *chatbot* fosse capaz de interpretar o desejo do usuário nas mensagens recebidas, conforme exibido na Figura 18.

Figura 18 – Aplicativo de conversa criado no LUIS



The screenshot shows the Microsoft LUIS portal interface. At the top, it says 'Cognitive Services | Language Understanding - West US'. Below that, it displays 'Conversation apps' and 'Azure subscription: Assinatura Gratuita / Authoring resource: luis-bot-tcc'. A table lists the conversation apps, with one entry: 'ModeloLuis' with a last modified date of '2/11/2020', culture 'pt-br', and 233 endpoint hits.

Name	Last modified ↓	Culture	Endpoint hits
ModeloLuis	2/11/2020	pt-br	233

Fonte: Autor (2020).

5.1.1.1 Cadastro das intenções no LUIS

Após criado o aplicativo no LUIS, foram cadastradas todas as intenções com exemplos de enunciados, também chamados de expressões ou declarações pela *Microsoft*, para treinamento do modelo personalizado, conforme comprovado na Figura 19 e detalhado no Quadro 7.

Figura 19 – Intenções cadastradas no LUIS

Name ↑	Examples	Features
ConsultarAntecipacao	15	+ Add feature
ConsultarAtraso	15	+ Add feature
ConsultarEstoque	15	+ Add feature
ConsultarPedido	15	+ Add feature
Cumprimentar	10	+ Add feature
Despedir	10	+ Add feature
None	10	+ Add feature

Fonte: Autor (2020).

Quadro 7 – Intenções e enunciados do *chatbot*

(continua)

Intenções	Enunciados
ConsultarAntecipacao	É possível antecipar a entrega do João?
	É possível antecipar a entrega da Maria?
	É possível antecipar a entrega do cliente Máquinas Sazi Ltda?
	É possível antecipar a entrega do cliente Máquinas Sazi Ltda para o dia 21/05/2022?
	É possível antecipar a entrega do cliente 35-1?
	É possível antecipar a entrega do cliente 35-1 para o dia 21/05/2022?
	É possível antecipar a entrega do item 99052356?
	É possível antecipar a entrega do item 99052356 para o dia 02/10/2020?
	É possível antecipar 1 peça do item 99052356?

(continuação)

Intenções	Enunciados
ConsultarAntecipacao	É possível antecipar a entrega de 1 peça do item 99052356 para o dia 21/07/2023?
	É possível antecipar a entrega do pedido 12345 para o dia 21/05/2022?
	É possível antecipar a entrega de 100 peças do pedido 12345 para o dia 27/03/2021?
	É possível antecipar a entrega do item 99052356 do pedido 12345 para o dia 13/12/2019?
	É possível antecipar a entrega de 100 peças do item 99052356 do pedido 12345 para o dia 10/02/2020?
	É possível antecipar a entrega da linha 7 do pedido 12345 para o dia 21/05/2022?
ConsultarAtraso	Quais pedidos de venda vão atrasar?
	Quais pedidos não serão entregues no prazo?
	Quais pedidos do João Silva irão atrasar?
	Quais pedidos do João atrasarão?
	Quais pedidos da Maria não serão entregues no prazo?
	Quais pedidos do cliente Máquinas Sazi Ltda irão atrasar?
	Quais pedidos do cliente 35-1 irão atrasar?
	O item 99052356 vai atrasar?
	O item 99052356 atrasará?
	Os pedidos do item 99052356 serão entregues no prazo?
	Quantos dias irá atrasar o pedido 12345?
	O pedido 12345 será entregue no prazo?
	O item 99052356 do pedido 12345 será entregue no prazo?
A linha 7 do pedido 12345 vai atrasar?	
A linha 7 do pedido 12345 será entregue no prazo?	
ConsultarEstoque	Quais pedidos já estão no estoque?
	Quais pedidos já estão prontos?
	Quais pedidos podem ser enviados para o cliente?
	O pedido do João Silva já está pronto?
	O pedido da Maria já foi produzido?
	Quais pedidos do cliente Máquinas Sazi Ltda já estão em estoque?
	Quais pedidos do cliente 35-1 já estão em estoque?
	O item 99052356 já está em estoque?
	Quantas peças do item 99052356 já estão no estoque?
	Os pedidos do item 99052356 já estão em estoque?
	O pedido 12345 já está em estoque?
	O pedido 12345 já está pronto?
	Qual é a quantidade em estoque do pedido 12345?
O item 99052356 do pedido 12345 já está em estoque?	
A linha 7 do pedido 12345 já está em estoque?	
ConsultarPedido	Me mostre as informações do pedido 12345.
	Quais são os pedidos pendentes do cliente Máquinas Sazi Ltda?

(continuação)

Intenções	Enunciados
ConsultarPedido	Quais são os pedidos pendentes do cliente 35-1?
	Quais são os pedidos pendentes do item 99052356?
	O item 99052356 está pendente?
	A linha 7 do pedido 12345 está pendente?
	Quais são os itens do pedido 12345?
	Quais são as linhas do pedido 12345?
	Qual é a quantidade do pedido 12345?
	Qual é a data de entrega original do pedido 12345?
	Qual é data de entrega reprogramada do pedido 12345?
	Qual é data de entrega original do item 99052356 do pedido 12345?
	Qual é data de entrega reprogramada do item 99052356 do pedido 12345?
	Qual é data de entrega original da linha 7 do pedido 12345?
	Qual é data de entrega reprogramada da linha 7 do pedido 12345?
	Cumprimentar
Olá, tudo bem?	
Tudo bem?	
Oi!	
Bom dia!	
Boa tarde!	
Boa noite!	
Opa!	
Buenas!	
Dae!	
Despedir	Tchau!
	Adeus!
	Até mais!
	Até logo!
	Obrigado!
	Muito obrigado!
	Valeu!
	Vlw!
	Flw!
Bom final de semana!	
None	Qual é o preço do item 99052356?
	Qual é o valor do item 99052356?
	Qual é o custo do item 99052356?
	A ordem 123456789 é necessária para atender a quais pedidos de venda?
	Aonde está a ordem 1234567?
	Qual é o próximo apontamento da ordem 987654321?
	Quem apontou a operação 242 da ordem 2548749?
	A ordem de produção 2548707 já foi entregue?
Quando foi entregue a ordem 2548709?	

(conclusão)

Intenções	Enunciados
None	Quem entregou a ordem 12345?

Fonte: Autor (2020).

A propósito, é importante citar que a intenção “None” é criada automaticamente em todos os modelos do LUIS e não pode ser excluída ou renomeada, porque é obrigatória, visto que é utilizada para ensinar ao LUIS quais declarações estão fora do domínio da aplicação, ou seja, tudo aquilo que o *chatbot* não está autorizado a responder para o usuário. Desse modo, foram apenas adicionados alguns enunciados de exemplo nessa intenção de *fallback*.

5.1.1.2 Cadastro das entidades no LUIS

Logo após o cadastro das intenções, todas as entidades especificadas na seção 4.3.4 que precisam ser reconhecidas pelo *chatbot* foram cadastradas no aplicativo do LUIS como sendo do tipo “*Machine learned*”, conforme demonstrado na Figura 20. Assim, de acordo com a documentação da *Microsoft* (2020b), o LUIS utilizará algoritmos de aprendizado de máquina para extrair as entidades dos enunciados a partir do contexto da conversa com o usuário.

Figura 20 – Entidades cadastradas no LUIS

Name ↑	Type
CodigoCliente	Machine learned
Data	Machine learned
Item	Machine learned
Linha	Machine learned
NomeCliente	Machine learned
Pedido	Machine learned
Quantidade	Machine learned

Fonte: Autor (2020).

5.1.1.3 Treinamento do modelo de linguagem

Depois de criado o aplicativo no LUIS, foi necessário treinar o modelo através do botão “*Train*” do portal do LUIS (MICROSOFT, 2020b), conforme indicado na Figura 21.

Figura 21 – Treinamento do modelo de linguagem no LUIS

The screenshot displays the LUIS portal interface. At the top, there are navigation tabs: DASHBOARD, BUILD, and MANAGE. The 'Train' button is highlighted with a red box. A notification box in the top right corner states 'Train completed' at 10:59 PM, with the message 'Training is up to date.' Below the notification, a table lists several intents. The table has three columns: Name, Examples, and Features. The intents listed are ConsultarAntecipacao, ConsultarAtraso, ConsultarEstoque, ConsultarPedido, Cumprimentar, Despedir, and None. Each intent has 15 examples and an 'Add feature' button.

Name ↑	Examples	Features
ConsultarAntecipacao	15	+ Add feature
ConsultarAtraso	15	+ Add feature
ConsultarEstoque	15	+ Add feature
ConsultarPedido	15	+ Add feature
Cumprimentar	10	+ Add feature
Despedir	10	+ Add feature
None	10	+ Add feature

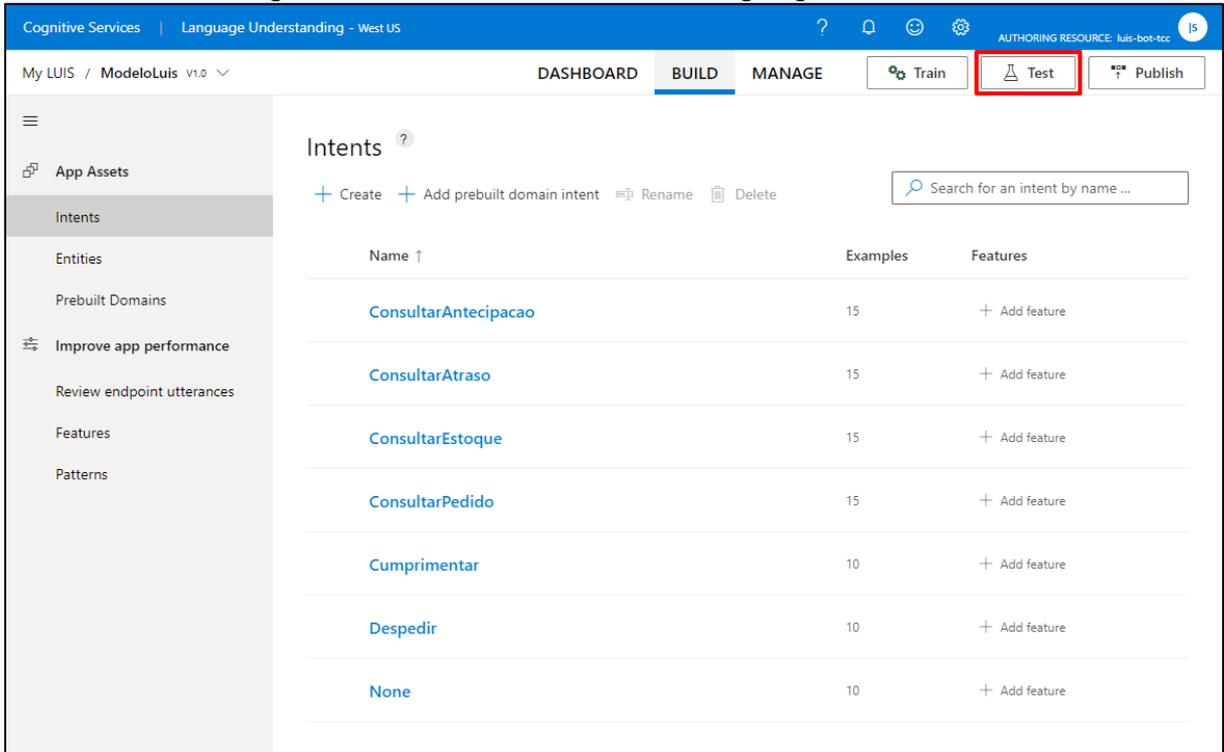
Fonte: Autor (2020).

De acordo com a documentação da *Microsoft* (2020b), o treinamento de um aplicativo do LUIS é um processo iterativo que serve para melhorar o reconhecimento da linguagem natural do usuário, e deve ser realizado sempre que uma intenção ou entidade for adicionada, alterada ou excluída do modelo.

5.1.1.4 Testes do modelo de linguagem

Imediatamente após cada treinamento do modelo, pelo menos um teste foi executado para garantir que o modelo continuava reconhecendo corretamente todas as intenções e entidades dos enunciados dos usuários. Os diversos testes unitários do aplicativo LUIS foram realizados por meio do botão “*Test*” do portal do LUIS (MICROSOFT, 2020b), conforme indicado na Figura 22 e exemplificado na Figura 23.

Figura 22 – Teste do modelo de linguagem no LUIS

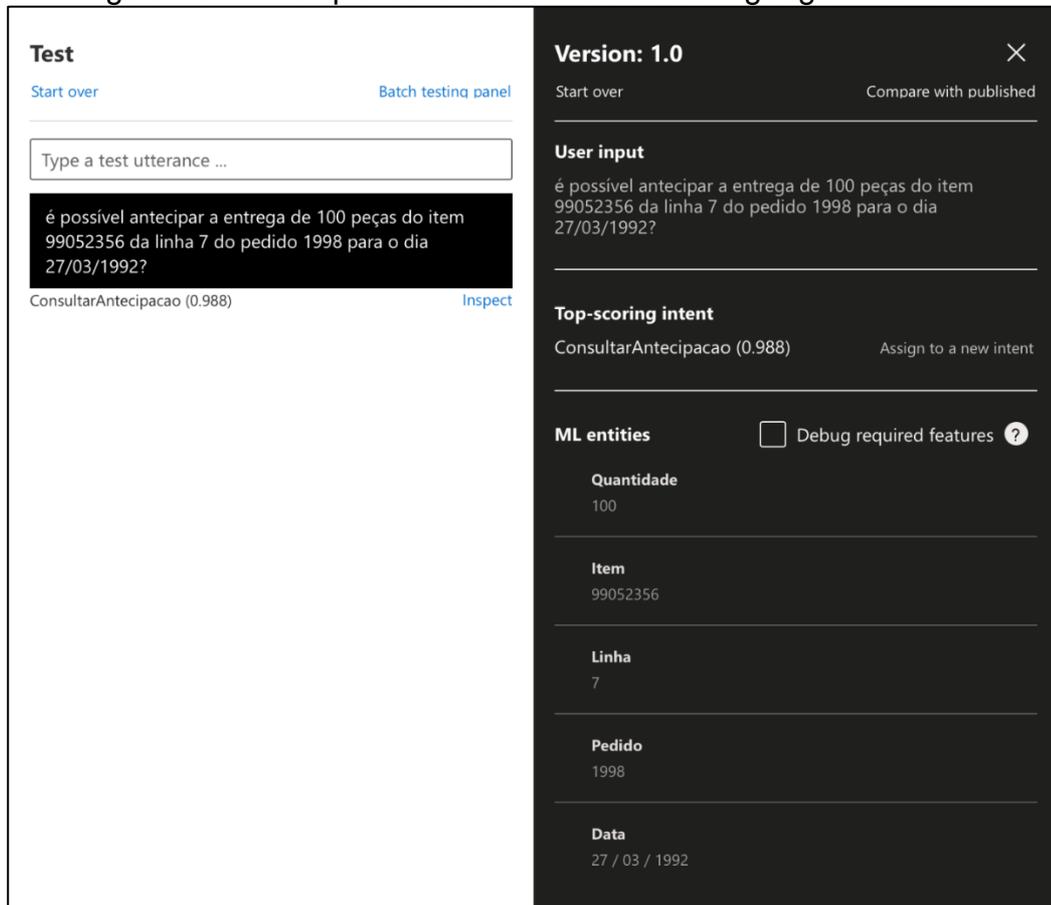


The screenshot shows the LUIS 'Intents' management page. The 'Test' button in the top navigation bar is highlighted with a red box. Below the navigation, there are tabs for 'DASHBOARD', 'BUILD', 'MANAGE', 'Train', 'Test', and 'Publish'. The main content area is titled 'Intents' and includes a search bar and action buttons like '+ Create', '+ Add prebuilt domain intent', 'Rename', and 'Delete'. A table lists the following intents:

Name ↑	Examples	Features
ConsultarAntecipacao	15	+ Add feature
ConsultarAtraso	15	+ Add feature
ConsultarEstoque	15	+ Add feature
ConsultarPedido	15	+ Add feature
Cumprimentar	10	+ Add feature
Despedir	10	+ Add feature
None	10	+ Add feature

Fonte: Autor (2020).

Figura 23 – Exemplo de teste do modelo de linguagem no LUIS



The screenshot displays the LUIS 'Test' interface. On the left, there is a text input field for the test utterance. The entered text is: "é possível antecipar a entrega de 100 peças do item 99052356 da linha 7 do pedido 1998 para o dia 27/03/1992?". Below the input, the system has identified the intent as "ConsultarAntecipacao" with a confidence score of 0.988. On the right, a detailed analysis panel shows the following information:

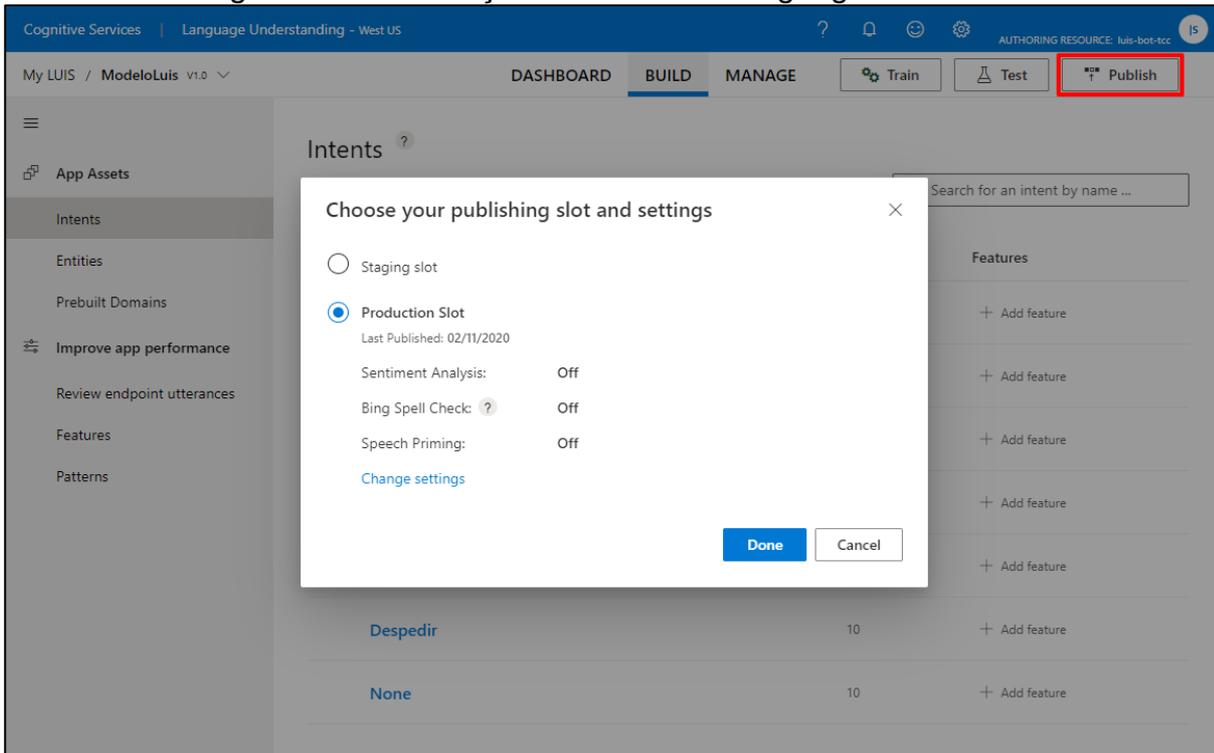
- Version:** 1.0
- User input:** é possível antecipar a entrega de 100 peças do item 99052356 da linha 7 do pedido 1998 para o dia 27/03/1992?
- Top-scoring intent:** ConsultarAntecipacao (0.988)
- ML entities:**
 - Quantidade: 100
 - Item: 99052356
 - Linha: 7
 - Pedido: 1998
 - Data: 27 / 03 / 1992

Fonte: Autor (2020).

5.1.1.5 Publicação do modelo de linguagem

Finalmente, depois de criar, treinar e testar o aplicativo no LUIS, o aplicativo foi publicado como um serviço *web* através do botão “*Publish*” do portal do LUIS (MICROSOFT, 2020b), conforme indicado na Figura 24.

Figura 24 – Publicação do modelo de linguagem no LUIS



Fonte: Autor (2020).

5.1.2 Criação de aplicativo no Azure

Com o aplicativo do LUIS publicado e disponível para consumo como um serviço *web*, o próximo passo foi criar o grupo de recursos “*rg-bot-tcc*” em uma assinatura gratuita do *Microsoft Azure* com a finalidade de armazenar todos os recursos da solução *web* que foi desenvolvida, conforme demonstrado na Figura 25.

Visando agilizar o processo de desenvolvimento do *chatbot*, foi utilizado um recurso do tipo “*web app bot*” que faz parte do Serviço de *Bot* do *Azure* (MICROSOFT, 2020a). Porém, é importante comentar que esse recurso foi criado selecionando a opção de linguagem de programação *C#* e o modelo “*Bot Básico*”, conforme indicado na Figura 26.

Figura 25 – Grupo de recursos criado no Azure

The screenshot shows the Azure portal interface for a resource group named 'rg-bot-tcc'. The top navigation bar includes the Microsoft Azure logo and a search bar. The left sidebar contains various navigation options like 'Visão geral', 'Log de atividade', 'IAM (Controle de acesso)', 'Marcações', 'Eventos', 'Configurações', and 'Gerenciamento de Custos'. The main content area displays the 'Fundamentos' section with details for 'Assinatura (alterar)' and 'Assinatura Gratuita'. Below this, there is a table of resources within the group, filtered by 'Tipo == tudo' and 'Localização == tudo'. The table lists six resources with their names, types, and locations.

Nome	Tipo	Localização
app-bot-tcc	Bot do Aplicativo Web	Global
app-bot-tcc	Serviço de Aplicativo	Oeste dos EUA
appi-bot-tcc	Application Insights	Oeste dos EUA
luis-bot-tcc	Serviços Cognitivos	Oeste dos EUA
luis-pred-bot-tcc	Serviços Cognitivos	Oeste dos EUA
plan-bot-tcc	Plano do Serviço de Aplicativo	Oeste dos EUA

Fonte: Autor (2020).

Figura 26 – Modelo do bot selecionado no Azure

The screenshot shows the 'Modelo de bot' selection screen in the Azure portal. The page title is 'Modelo de bot' and it prompts the user to 'Escolha um modelo'. Under the heading 'Selecionar um modelo de bot', there is a note that bot models include source code and services like 'Reconhecimento Vocal' and 'Análise de Bot'. The 'Idioma do SDK' is set to 'C#'. There are two main options for bot models: 'Echo Bot' (C#, simple bot) and 'Bot Básico' (C#, includes 'Reconhecimento vocal' and 'Análise de Bot'). Below these, there are sections for 'Assistente Virtual', 'Bot de Reconhecimento Vocal', and 'Bot QnA', each with a brief description of what they include.

Fonte: Autor (2020).

Dessa maneira, não foi necessário desenvolver toda a aplicação do zero, pois foi utilizado um programa modelo disponibilizado gratuitamente pela *Microsoft*.

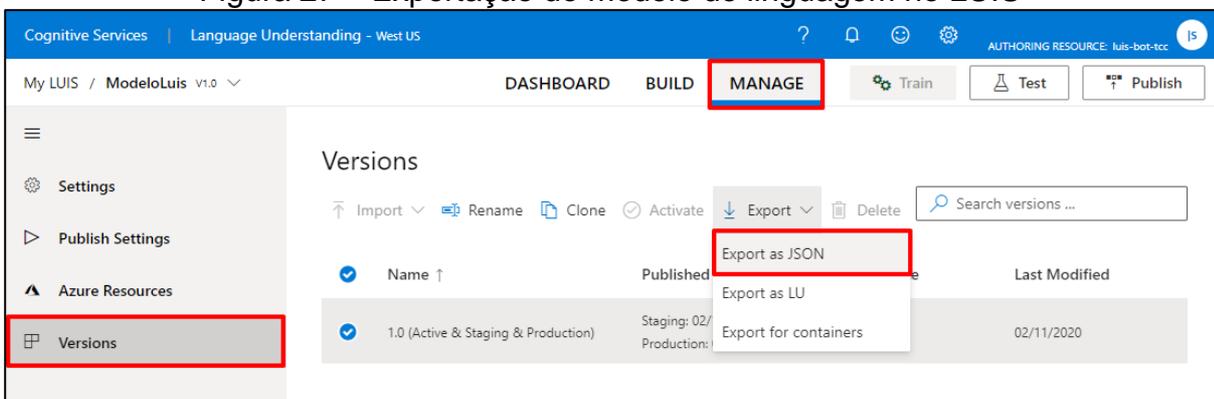
Entretanto, mesmo assim, diversas modificações foram necessárias no código-fonte do programa modelo para atender a todos os requisitos deste trabalho.

5.1.2.1 Integração do chatbot com o LUIS

A primeira modificação do código-fonte do modelo de *bot* da *Microsoft* foi a integração com o aplicativo publicado no LUIS especificamente para interpretar as intenções e entidades que fazem parte do escopo deste projeto, e para isso foi necessário apenas:

- exportar o modelo de linguagem no formato JSON através do portal do LUIS (MICROSOFT, 2020b), conforme indicado na Figura 27;
- gerar uma classe C# para o modelo de linguagem exportado utilizando a ferramenta *LUISGen* da *Microsoft* e o comando destacado na Figura 28.

Figura 27 – Exportação do modelo de linguagem no LUIS



Fonte: Autor (2020).

Figura 28 – Comando para geração de classe C# do modelo de linguagem

```

Administrador: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> cd C:\Users\amamaro\source\app-bot-tcc-src\CognitiveModels
PS C:\Users\amamaro\source\app-bot-tcc-src\CognitiveModels> luisgen ModeloLuis.json -cs
Generating file ModeloLuis.cs that contains class Luis.ModeloLuis.
PS C:\Users\amamaro\source\app-bot-tcc-src\CognitiveModels> exit

```

Fonte: Autor (2020).

5.1.2.2 Integração do chatbot com o Drummer APS

Com o objetivo de atender aos requisitos RF02, RF03, RF04, RF05 e RNF01, o *chatbot* desenvolvido neste trabalho foi integrado ao sistema *Drummer* APS, que já era utilizado pela organização em questão.

Contudo, como o fornecedor do sistema *Drummer* APS não disponibiliza o recurso de integração por API, que é o método de integração recomendado para esse caso, foi preciso utilizar as técnicas de acesso direto à base de dados e execução por linha de comando para integrar o *chatbot* com o SI.

5.1.2.2.1 Acesso direto à base de dados

O primeiro passo para integrar os sistemas foi cadastrar o usuário “BOT” no ambiente “DRUMMER8” do *Drummer* APS com as mesmas permissões de acesso de um planejador mestre, atendendo ao requisito RNF06.

Logo após, foi atendido o requisito RNF07 mediante a criação de um usuário no SGBD *Oracle* da Sazi com o nome “DRUMMER8_BOT”, a partir de um clone do usuário “DRUMMER8”, que é o usuário padrão do aplicativo. Nesse ponto, é importante explicar que todos os usuários do banco de dados cujo nome inicia com a palavra “DRUMMER8” são automaticamente considerados ambientes distintos pelo sistema *Drummer* APS.

Figura 29 –Tela de login do *Drummer* APS



Drummer Login

User:

Password:

Environment: ▼

Server: ▼

 **LINTER**

Copyright © 1999-2019 Linter Sistemas Ltda
All rights reserved. Linter, Linter's logo and Drummer are registered
trademarks of Linter Sistemas Ltda.
All other marks are property of their respective owners.

Fonte: Autor (2020).

Dessa maneira, o *chatbot* acessará somente o ambiente “DRUMMER8_BOT” do sistema *Drummer APS*, utilizando o usuário “BOT”, tal como exemplificado na tela de login da Figura 29. Assim, o *chatbot* poderá simular antecipações de pedidos de venda no ambiente “DRUMMER8_BOT” sem causar danos nos dados oficiais do ambiente de produção, denominado “DRUMMER8”.

A visão “VPEDIDOS” também foi criada no esquema “DRUMMER8_BOT” do SGBD *Oracle* para normalizar a consulta das informações dos pedidos de venda cadastrados no *Drummer APS*. O comando SQL utilizado para criar a visão “VPEDIDOS” pode ser conferido na Figura 30, e na Figura 31 é possível observar um exemplo de como poderiam ser consultados todos os pedidos de venda cadastrados no *Drummer APS* utilizando essa nova visão.

Figura 30 – Visão de pedidos de venda criada no *Drummer APS*

```

1  • CREATE OR REPLACE VIEW DRUMMER8_BOT.VPEDIDOS AS
2  SELECT nec.nped AS num_pedido
3         , nec.iped AS linha
4         , nec.ccli AS cod_cliente
5         , cli.razs AS nome_cliente
6         , nec.citm AS cod_item
7         , itm.ditm AS desc_item
8         , nec.dorg AS data_original
9         , nec.dtnc AS data_reprogramada
10        , TRUNC(nec.dtnc - nec.dorg) AS atraso
11        , nec.qtnc AS qtde_necessaria
12        , nec.qtrq AS qtde_requisitada
13        , nec.qtnc - nec.qtrq AS qtde_pendente
14        , NVL( (SELECT SUM(log.qtnc)
15                FROM dbordlog log
16                WHERE log.nec_id = nec.nec_id
17                    AND log.folg = 'E')
18              , 0
19              ) AS qtde_estoque
20  FROM dbfunnec nec
21  JOIN drfuncli cli ON cli.ccli = nec.ccli
22  JOIN dbfunitm itm ON itm.citm = nec.citm
23  WHERE nec.creg = 'P'
24  ORDER BY nec.nped
25         , nec.iped;

```

Fonte: Autor (2020).

Figura 31 – Exemplo de consulta de pedidos de venda no *Drummer* APS

```

1 SELECT *
2 FROM vpedidos;
    
```

NUM_PEDIDO	LINHA	COD_CLIENTE	NOME_CLIENTE	COD_ITEM	DESC_ITEM	DATA_ORIGINAL	DATA_REPROGRAMADA	ATRASSO	QTDE_NECESSARIA	QTDE_REQUISITADA	QTDE_PENDENTE	QTDE_ESTOQUE
17541	1	307-1	CONFIDENCIAL	10050	CONFIDENCIAL	01/12/2020	16/12/2020	15	10	0	10	10
17544	1	1170-1	CONFIDENCIAL	319517	CONFIDENCIAL	23/10/2020	16/12/2020	54	16	0	16	16
17553	9	38-1	CONFIDENCIAL	99051102	CONFIDENCIAL	11/12/2019	21/01/2021 08:57:00	407	102	0	102	0
17553	10	38-1	CONFIDENCIAL	308487	CONFIDENCIAL	08/12/2019	20/01/2021 15:08:00	409	2	0	2	0
17554	22	38-1	CONFIDENCIAL	313815	CONFIDENCIAL	03/12/2020	20/01/2021 16:40:00	48	20	0	20	0
17554	24	38-1	CONFIDENCIAL	99051059	CONFIDENCIAL	08/12/2019	22/01/2021 14:35:00	411	1	0	1	0
17621	1	38-3	CONFIDENCIAL	99061703	CONFIDENCIAL	30/11/2020	16/12/2020	16	15	0	15	15
17621	3	38-3	CONFIDENCIAL	99040911	CONFIDENCIAL	23/11/2019	16/12/2020	389	5	0	5	5
17642	2	38-2	CONFIDENCIAL	99035822	CONFIDENCIAL	07/12/2019	14/01/2021 21:56:00	404	80	56	24	0
17647	8	38-2	CONFIDENCIAL	99042935	CONFIDENCIAL	15/12/2020	18/01/2021 08:03:00	34	100	99	1	0
17647	28	38-2	CONFIDENCIAL	99030915	CONFIDENCIAL	09/12/2019	16/12/2020	373	131	0	131	131
17647	29	38-2	CONFIDENCIAL	99027654	CONFIDENCIAL	26/11/2019	16/12/2020	386	88	0	88	0
17647	33	38-2	CONFIDENCIAL	99029057	CONFIDENCIAL	15/12/2020	21/01/2021 09:58:00	37	90	86	4	0
17647	52	38-2	CONFIDENCIAL	99053038	CONFIDENCIAL	03/12/2019	19/01/2021	413	25	0	25	0
17647	56	38-2	CONFIDENCIAL	99053039	CONFIDENCIAL	11/12/2019	19/01/2021	405	44	0	44	0

Fonte: Autor (2020).

Sendo assim, para que o *chatbot* consiga acessar a base de dados e possa consultar a visão “VPEDIDOS” do ambiente “DRUMMER8_BOT” do SI *Drummer* APS, foi criada a classe “*DBConnection*” que encapsula a *string* de conexão com o banco de dados *Oracle* da Sazi. Vide o código-fonte da classe “*DBConnection*” com dados de conexão fictícios na Figura 32.

Figura 32 – Classe *DBConnection*

```

1 namespace CoreBot
2 {
3     public static class DBConnection
4     {
5         private static readonly string conString = "User Id=DRUMMER8_BOT;Password=password;Data Source=192.168.21.0:1521/PRO.COM.BR;";
6
7         public static string ConString => conString;
8     }
9 }
    
```

Fonte: Autor (2020).

5.1.2.2.2 Execução por linha de comando

Apesar do *chatbot* já ter acesso ao banco de dados *Oracle* do sistema *Drummer* APS, ainda não era possível responder algumas das principais perguntas sobre pedidos de venda realizadas pelos usuários, que são aquelas associadas à antecipação de entrega de um determinado pedido de venda.

Pois bem, para responder a essas perguntas mais complexas, é indispensável que o *chatbot* execute uma simulação no sistema *Drummer* APS para validar a antecipação de entrega de um determinado pedido de venda, considerando a demanda, a capacidade finita dos recursos produtivos e todos os demais parâmetros que são cadastrados apenas no sistema APS.

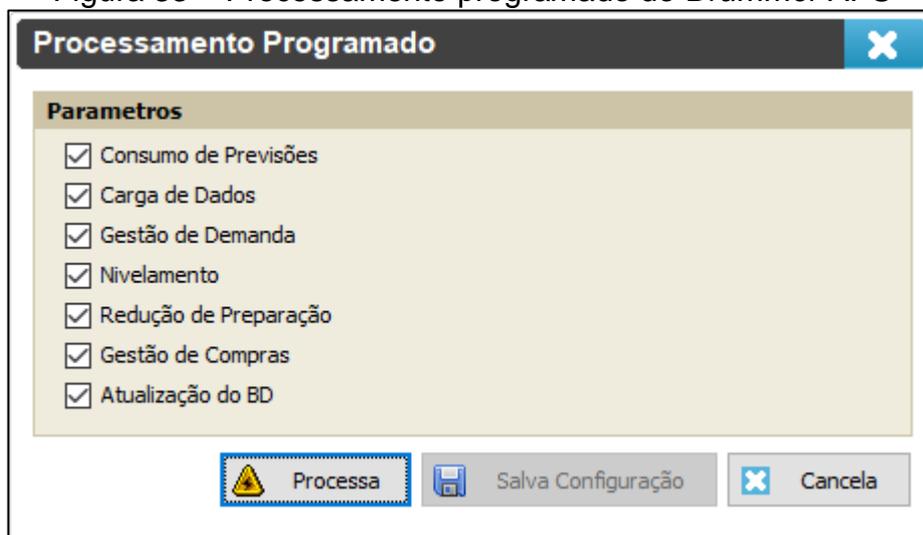
Para realizar uma simulação desse tipo é necessário alterar a data de entrega original, e a quantidade se for uma antecipação parcial, do pedido de venda informado pelo usuário no *Drummer* APS, executar um processamento completo no ambiente “DRUMMER8_BOT” e verificar se o sistema calculou atraso para o pedido em questão. Nesse caso, se o sistema *Drummer* APS indicar atraso na entrega, o *chatbot* deve responder para o usuário que não é possível antecipar a entrega do pedido de venda. Por outro lado, se não houver atraso, o *chatbot* pode autorizar a antecipação do pedido.

O processamento completo do *Drummer* APS envolve a execução das seguintes transações no sistema:

- a) consumo de previsões (F2);
- b) carga de dados (F3);
- c) gestão de demanda (F4);
- d) nivelamento (F5);
- e) redução de preparação (F6);
- f) gestão de compras (F7);
- g) atualização do BD (F8).

Cada uma dessas transações pode ser executada individualmente pelo usuário, ou em sequência pelo sistema através da função de processamento programado, que pode ser acionada pela tecla de atalho F9 e parametrizada na tela exibida na Figura 33.

Figura 33 – Processamento programado do *Drummer* APS



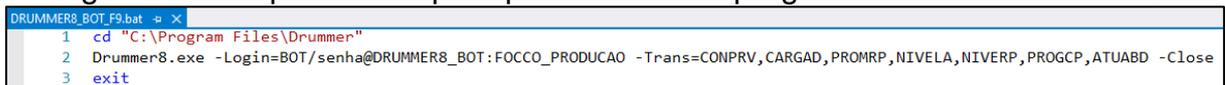
Fonte: Autor (2020).

Entretanto, existem apenas três formas de executar essas transações no sistema *Drummer* APS, conforme listado a seguir:

- a) pelo menu de opções do sistema;
- b) pelas respectivas teclas de função (F2 até F9);
- c) por linha de comando.

Desse modo, especificamente para atender ao requisito RF05 deste projeto, optou-se por automatizar o processamento programado do *Drummer* APS por meio da execução de todas as transações listadas anteriormente por linha de comando. Sendo assim, o arquivo em lotes “DRUMMER8_BOT_F9” foi desenvolvido com os comandos mostrados na Figura 34 e adicionado ao projeto do *chatbot* com as propriedades indicadas na Figura 35.

Figura 34 – Arquivo *batch* para processamento programado do *Drummer* APS

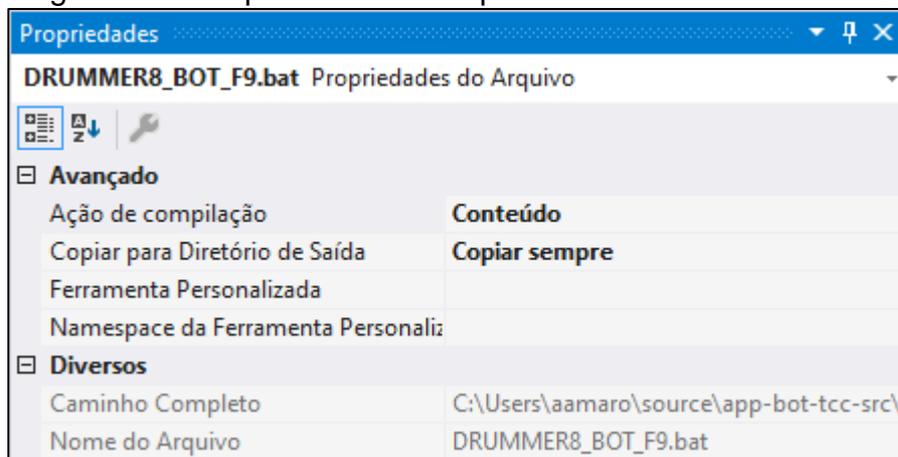


```

DRUMMER8_BOT_F9.bat + x
1 cd "C:\Program Files\Drummer"
2 Drummer8.exe -Login=BOT/senha@DRUMMER8_BOT:FOCCO_PRODUCAO -Trans=CONPRV,CARGAD,PROMRP,NIVELA,NIVERP,PROGCP,ATUABD -Close
3 exit
  
```

Fonte: Autor (2020).

Figura 35 – Propriedades do arquivo *batch* do *Drummer* APS



Fonte: Autor (2020).

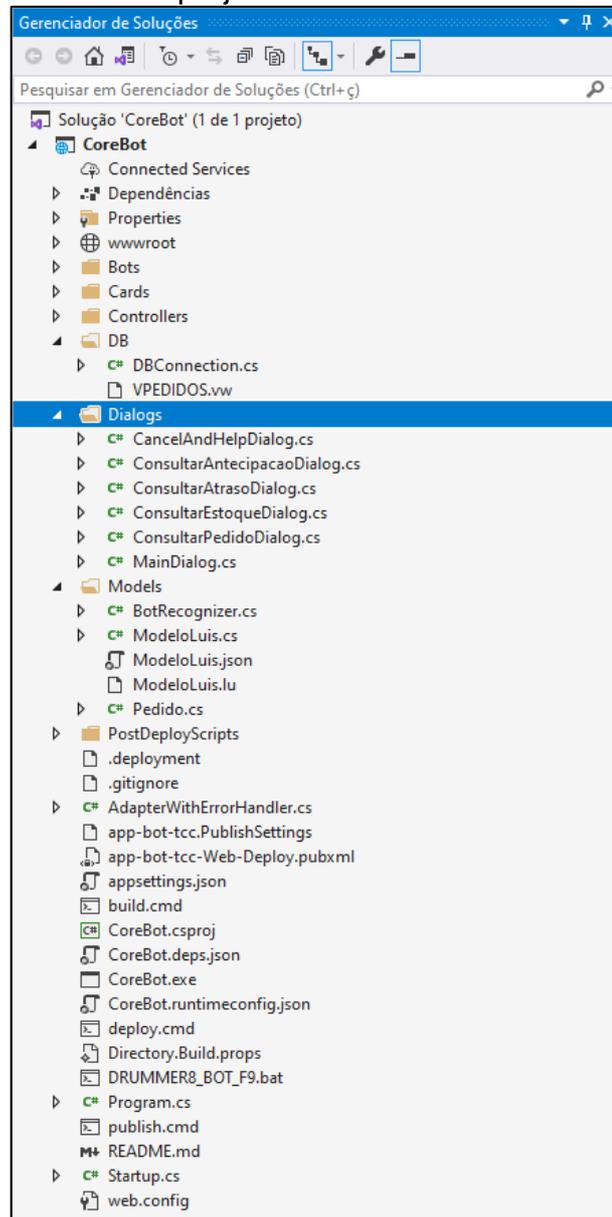
A linha de comando utilizada no arquivo *batch* ilustrado na Figura 34 abrirá localmente o *software* *Drummer* APS, acessará o sistema com o usuário “BOT” no ambiente “DRUMMER8_BOT” do servidor “FOCCO_PRODUCAO”, executará todas as transações do processamento programado em sequência e fechará o aplicativo automaticamente.

5.1.2.3 Codificação do chatbot

As classes do projeto do *chatbot* foram organizadas no formato de diálogos, onde foi criada uma classe para conduzir o diálogo de cada uma das principais intenções do usuário, que foram especificadas no Quadro 6, conforme pode ser observado na Figura 36.

Dessa maneira, quando o usuário envia uma nova mensagem para o *chatbot*, seja ela qual for, é instanciado um novo objeto da classe “*MainDialog*”, que é a classe responsável por controlar todo o fluxo da conversa com o usuário.

Figura 36 – Estrutura do projeto do *chatbot* no *Microsoft Visual Studio*



Fonte: Autor (2020).

Imediatamente após receber a primeira mensagem do usuário, o método “*IntroStepAsync*” da classe “*MainDialog*” retorna uma mensagem para o usuário se apresentando, deixando claro para a pessoa que ela está falando com um robô, e esclarecendo quais tipos de pergunta que o *chatbot* é capaz de responder, conforme detalhado no código apresentado na Figura 37.

Figura 37 – Método *IntroStepAsync* da classe *MainDialog*

```
private async Task<DialogTurnResult> IntroStepAsync(WaterfallStepContext stepContext, CancellationToken cancellationToken)
{
    if (!_luisRecognizer.IsConfigured)
    {
        await stepContext.Context.SendActivityAsync(MessageFactory.Text("ATENÇÃO: O LUIS não está configurado.",
            , inputHint: InputHints.IgnoringInput), cancellationToken);
        return await stepContext.NextAsync(null, cancellationToken);
    }

    var messageText = stepContext.Options?.ToString() ?? "Olá!\n\n"
        + "Meu nome é TCC e eu sou um bot que sabe tudo sobre pedidos de venda.\n\nComo eu posso lhe ajudar?";
    var promptMessage = MessageFactory.Text(messageText, messageText, InputHints.ExpectingInput);
    return await stepContext.PromptAsync(nameof(TextPrompt), new PromptOptions{Prompt = promptMessage}, cancellationToken);
}
```

Fonte: Autor (2020).

A partir da segunda mensagem recebida do usuário, o método “*ActStepAsync*” da classe “*MainDialog*” utiliza o modelo de linguagem natural que foi publicado no LUIS para interpretar qual é a intenção mais provável do usuário com aquele enunciado e extrair todas as entidades contidas nele.

As entidades extraídas pelo LUIS do enunciado são utilizadas para instanciar um objeto da classe “*Pedido*” que será mantido, e possivelmente atualizado, até o encerramento do diálogo. Vide as propriedades da classe “*Pedido*” na Figura 38.

Figura 38 – Classe *Pedido*

```
1 namespace CoreBot
2 {
3     public class Pedido
4     {
5         public string Numero { get; set; }
6
7         public string Linha { get; set; }
8
9         public string Item { get; set; }
10
11        public string CodigoCliente { get; set; }
12
13        public string NomeCliente { get; set; }
14
15        public string Quantidade { get; set; }
16
17        public string Data { get; set; }
18    }
19 }
```

Fonte: Autor (2020).

Já a mais provável intenção identificada pelo LUIS é utilizada para orientar o fluxo da conversa, conforme pode ser observado no código-fonte apresentado na Figura 39.

Figura 39 – Método *ActStepAsync* da classe *MainDialog*

```
private async Task<DialogTurnResult> ActStepAsync(WaterfallStepContext stepContext, CancellationToken cancellationToken)
{
    if (!_luisRecognizer.IsConfigured)
    {
        var luisResult = await _luisRecognizer.RecognizeAsync<ModeloLuis>(stepContext.Context, cancellationToken);
        var pedido = new Pedido()
        {
            Numero = luisResult.Entities.Pedido?.FirstOrDefault(),
            Linha = luisResult.Entities.Linha?.FirstOrDefault(),
            Item = luisResult.Entities.Item?.FirstOrDefault(),
            CodigoCliente = luisResult.Entities.CodigoCliente?.FirstOrDefault(),
            NomeCliente = luisResult.Entities.NomeCliente?.FirstOrDefault(),
            Quantidade = luisResult.Entities.Quantidade?.FirstOrDefault(),
            Data = luisResult.Entities.Data?.FirstOrDefault(),
        };
        switch (luisResult.TopIntent().intent)
        {
            case ModeloLuis.Intent.Cumprimentar:
                var promptMessage = "Olá!\n\nComo eu posso lhe ajudar?";
                return await stepContext.ReplaceDialogAsync(InitialDialogId, promptMessage, cancellationToken);

            case ModeloLuis.Intent.ConsultarAntecipacao:
                return await stepContext.BeginDialogAsync(nameof(ConsultarAntecipacaoDialog), pedido, cancellationToken);

            case ModeloLuis.Intent.ConsultarAtraso:
                return await stepContext.BeginDialogAsync(nameof(ConsultarAtrasoDialog), pedido, cancellationToken);

            case ModeloLuis.Intent.ConsultarEstoque:
                return await stepContext.BeginDialogAsync(nameof(ConsultarEstoqueDialog), pedido, cancellationToken);

            case ModeloLuis.Intent.ConsultarPedido:
                return await stepContext.BeginDialogAsync(nameof(ConsultarPedidoDialog), pedido, cancellationToken);

            case ModeloLuis.Intent.Despedir:
                var byeMessageText = $"Espero que eu tenha esclarecido a sua dúvida.\n\nAté a próxima!";
                var byeMessage = MessageFactory.Text(byeMessageText, byeMessageText, InputHints.IgnoringInput);
                await stepContext.Context.SendActivityAsync(byeMessage, cancellationToken);
                return await stepContext.EndDialogAsync(null, cancellationToken);

            case ModeloLuis.Intent.None:
                var noneMessageText = $"Desculpa, mas eu não posso lhe responder isso.";
                var noneMessage = MessageFactory.Text(noneMessageText, noneMessageText, InputHints.IgnoringInput);
                await stepContext.Context.SendActivityAsync(noneMessage, cancellationToken);
                break;

            default:
                var understandMessageText = $"Desculpa, mas eu não entendi o que você quis dizer.\n\n"
                    + $"Por favor, tente perguntar de uma maneira diferente.";
                var understandMessage = MessageFactory.Text(understandMessageText, understandMessageText,
                    InputHints.IgnoringInput);
                await stepContext.Context.SendActivityAsync(understandMessage, cancellationToken);
                break;
        }
    }
    return await stepContext.NextAsync(null, cancellationToken);
}
```

Fonte: Autor (2020).

Ao final do diálogo, o método "*FinalStepAsync*" da classe "*MainDialog*" é executado e o *bot* questiona se o usuário gostaria de continuar a conversa realizando uma nova pergunta, conforme código-fonte apresentado na Figura 40. Caso o usuário

responda com uma mensagem qualquer, um novo diálogo é iniciado a partir do método “*ActStepAsync*” da mesma classe.

Figura 40 – Método *FinalStepAsync* da classe *MainDialog*

```
private async Task<DialogTurnResult> FinalStepAsync(WaterfallStepContext stepContext, CancellationToken cancellationToken)
{
    var promptMessage = "O que mais eu posso fazer por você?";
    return await stepContext.ReplaceDialogAsync(InitialDialogId, promptMessage, cancellationToken);
}
```

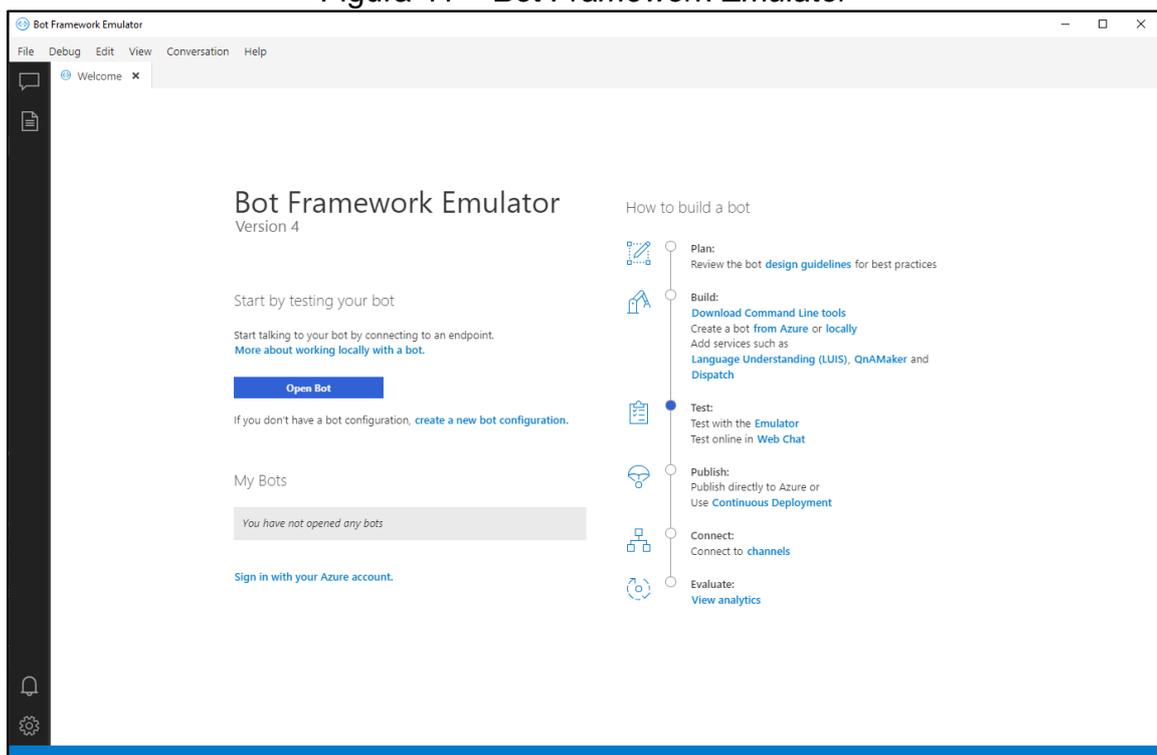
Fonte: Autor (2020).

5.2 TESTES

Ainda durante a etapa anterior, foram executados diversos testes unitários da aplicação que estava sendo codificada. Porém, ao final da etapa de desenvolvimento, uma versão *alfa* do *chatbot* foi submetida a testes mais rigorosos de funcionalidade, usabilidade, navegabilidade, estabilidade, integração, desempenho e segurança, pois essas características são primordiais para garantir um *software* de qualidade, de acordo com a norma ABNT NBR ISO/IEC 9126-1:2003.

Nessa etapa foi utilizado o aplicativo *desktop Bot Framework Emulator* da *Microsoft*, exibido na Figura 41, para testar e depurar o *chatbot* desenvolvido.

Figura 41 – *Bot Framework Emulator*



Fonte: Autor (2020).

Basicamente, após executar o programa localmente em modo *debug* pelo IDE *Microsoft Visual Studio*, bastava conectar-se ao *endpoint* da aplicação através do botão “*Open Bot*” do aplicativo *Bot Framework Emulator*, conforme demonstrado na Figura 42, e executar os testes do sistema, tal como exemplificado na Figura 43.

Figura 42 – Tela de conexão do *Bot Framework Emulator*

Fonte: Autor (2020).

Figura 43 – Exemplo de teste no *Bot Framework Emulator*

Fonte: Autor (2020).

Ao final dessa etapa, todas as eventuais falhas identificadas no sistema foram corrigidas, de modo a possibilitar a geração de uma versão *closed beta* da aplicação para ser utilizada pelos usuários chave nos pilotos.

5.3 IMPLANTAÇÃO

Após desenvolvido e testado o *chatbot*, a aplicação *web* foi implantada no Azure pela interface de linha de comando (ou CLI, do termo em inglês *Command-Line Interface*), conforme demonstrado na Figura 44.

Figura 44 – Implantação do *chatbot* pela CLI do Azure

```

Administrador: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> az webapp deployment source config-zip --resource-group "rg-bot-tcc" --name "app-bot-tcc"
--src "C:\app-bot-tcc-src.zip"
Getting scm site credentials for zip deployment
Starting zip deployment. This operation can take a while to complete ...
Deployment endpoint responded with status code 202
{
  "active": true,
  "author": "N/A",
  "author_email": "N/A",
  "complete": true,
  "deployer": "ZipDeploy",
  "end_time": "2020-12-18T03:43:36.234414Z",
  "id": "a8dd80a421ef4e6eadf1848ca882257f",
  "is_readonly": true,
  "is_temp": false,
  "last_success_end_time": "2020-12-18T03:43:36.234414Z",
  "log_url": "https://app-bot-tcc.scm.azurewebsites.net/api/deployments/latest/log",
  "message": "Created via a push deployment",
  "progress": "",
  "provisioningState": null,
  "received_time": "2020-12-18T03:43:14.5552967Z",
  "site_name": "app-bot-tcc",
  "start_time": "2020-12-18T03:43:14.8834114Z",
  "status": 4,
  "status_text": "",
  "url": "https://app-bot-tcc.scm.azurewebsites.net/api/deployments/latest"
}
PS C:\WINDOWS\system32> exit

```

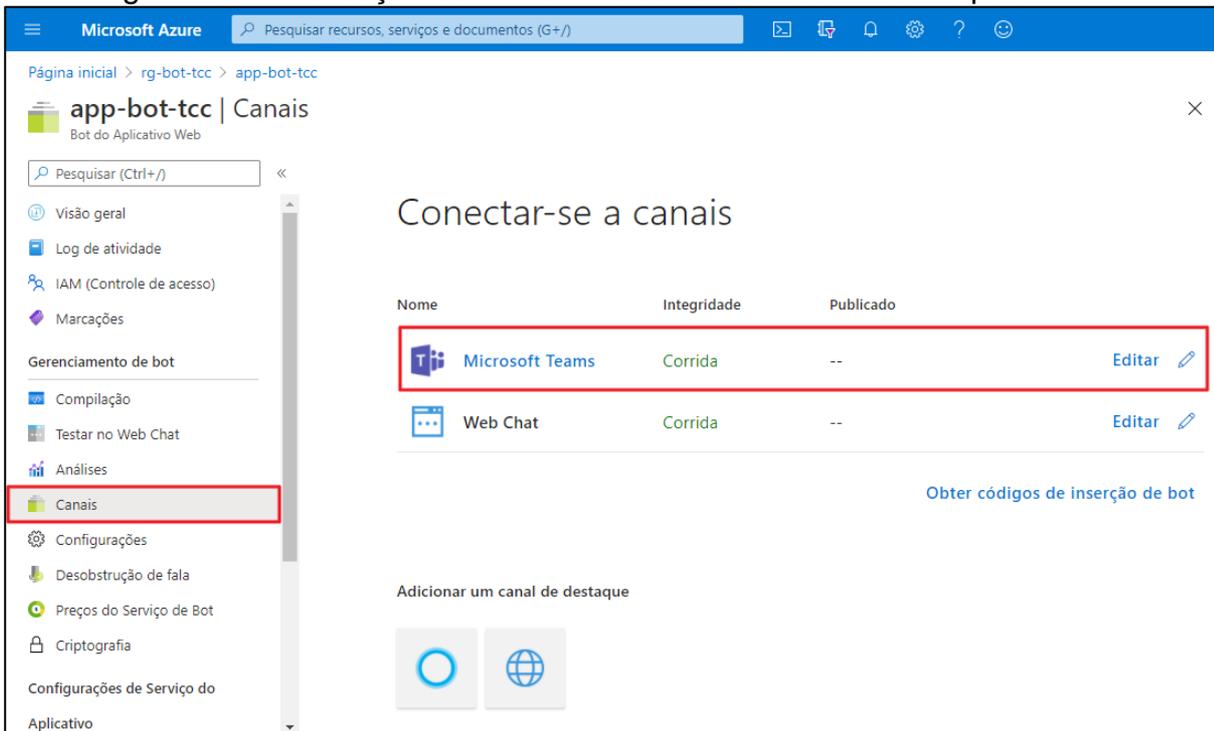
Fonte: Autor (2020).

5.4 PUBLICAÇÃO

O processo de publicação do *chatbot* no canal *Microsoft Teams* foi facilitado devido ao uso do *Azure Bot Service* da *Microsoft* (2020a), porque foi necessário apenas conectar o aplicativo *web* “app-bot-tcc” ao canal *Microsoft Teams* na guia “Canais” do portal do Azure, conforme indicado na Figura 45.

Assim que o aplicativo foi conectado ao canal, o *bot* já pôde ser adicionado aos *chats* do *Teams* através do código de inserção obtido na mesma guia do Azure.

Figura 45 – Publicação do *chatbot* no canal *Microsoft Teams* pelo *Azure*



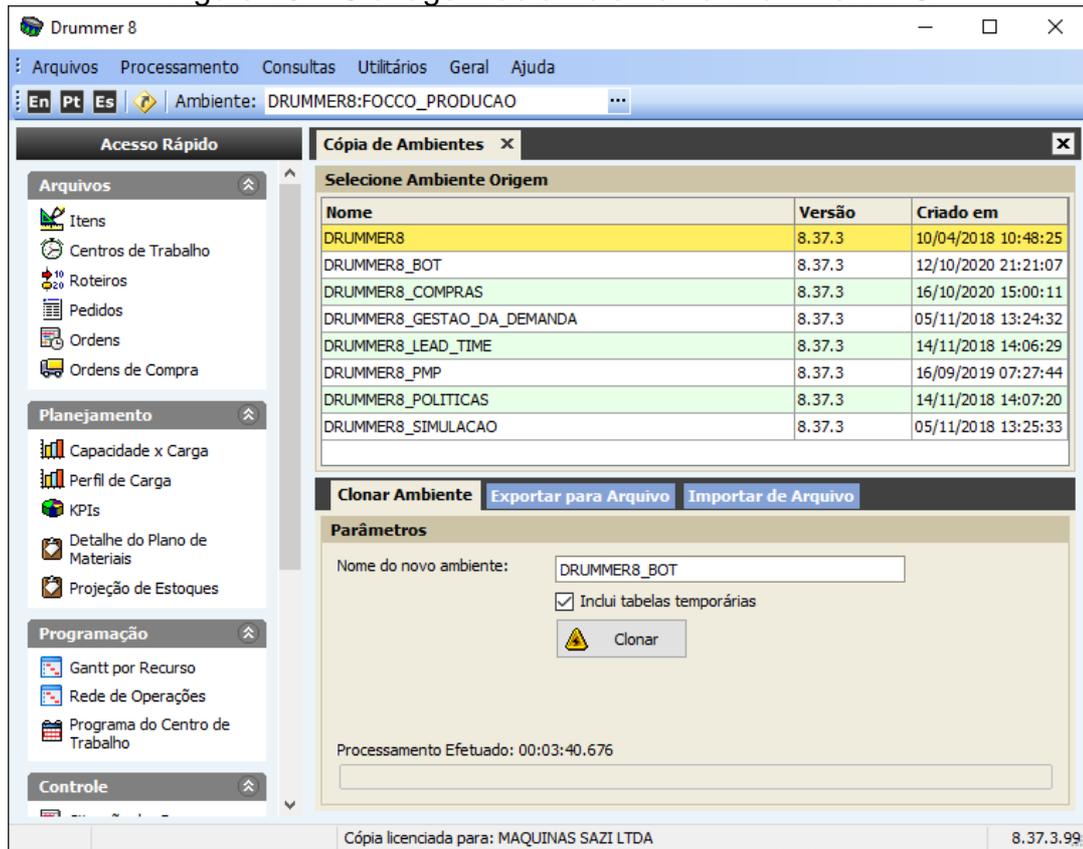
Fonte: Autor (2020).

5.5 PILOTO

Depois de publicada a versão *closed beta* do *chatbot* no canal *Microsoft Teams*, foi realizado um piloto com cada um dos usuários chave para validar a solução implementada, utilizando o método científico da experimentação e a técnica de observação estruturada para coletar dados referente a aderência do produto para os processos aos quais o sistema se predispôs a apoiar (SOMMERVILLE, 2018).

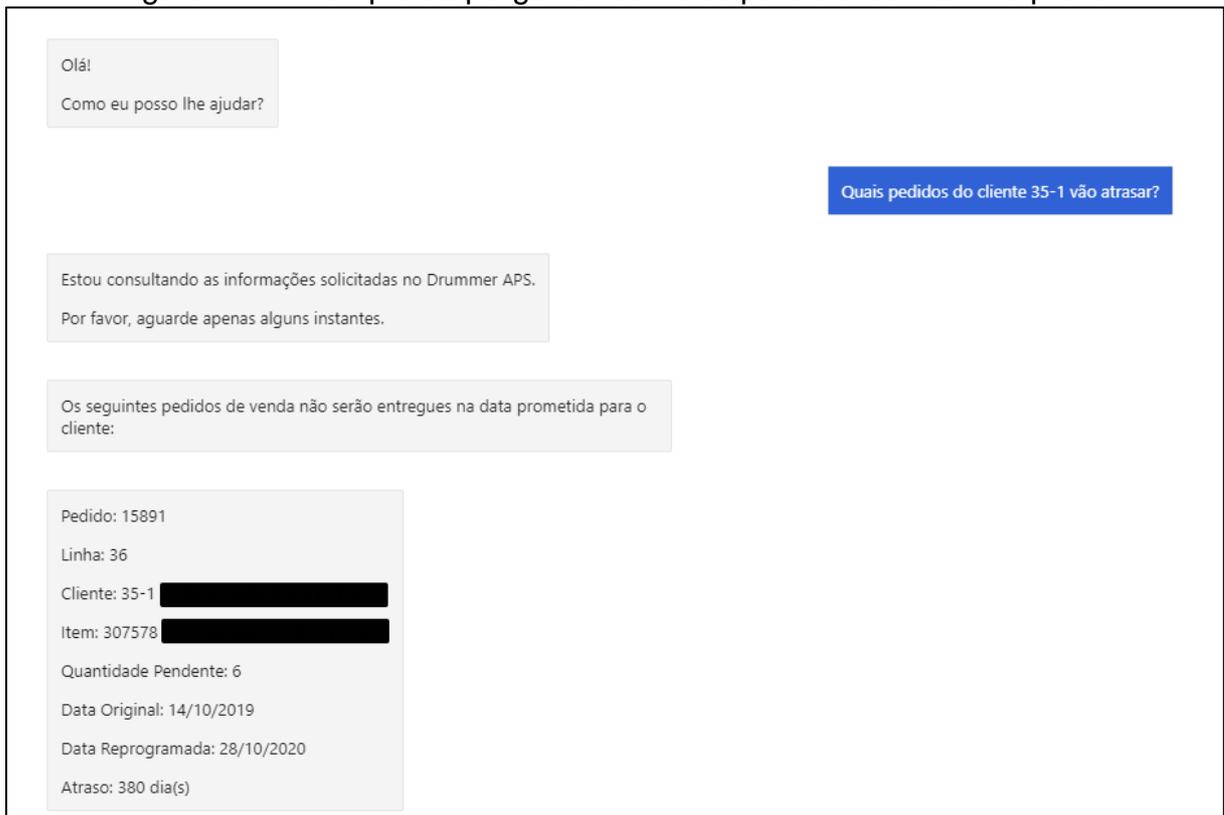
Antes de iniciar cada piloto, foi utilizada a funcionalidade de clonagem de ambientes do sistema *Drummer APS* para substituir todos os dados do ambiente de homologação “DRUMMER8_BOT” pelos dados do ambiente de produção “DRUMMER8”, conforme demonstrado na Figura 46.

Durante os pilotos, foi solicitado que os usuários chave realizassem uma série de perguntas distintas para o *chatbot* sobre os pedidos de venda da empresa, com base nos questionamentos que eles realmente haviam respondido nos últimos dias. A Figura 47 mostra uma das perguntas que foi realizada pelo usuário chave do setor de PCP no piloto, enquanto que a Figura 48 exemplifica o piloto com uma das perguntas que foi realizada pelo usuário chave da área comercial.

Figura 46 – Clonagem de ambiente no *Drummer APS*

Fonte: Autor (2020).

Figura 47 – Exemplo de pergunta realizada pelo UC do PCP no piloto



Fonte: Autor (2020).

Figura 48 – Exemplo de pergunta realizada pelo UC do Comercial no piloto

Olá!
Como eu posso lhe ajudar?

É possível antecipar 20 peças do cliente 4-1

O cliente 4-1 possui 6 pedidos de venda pendentes.
Por favor, informe o número do pedido que deseja antecipar.

17926

O pedido 17926 possui 3 linhas.
Por favor, informe o código do item que deseja antecipar.

309990

Para qual data você deseja antecipar a linha 2 do pedido 17926?

08/11/2020

Estou simulando a antecipação da linha 2 do pedido 17926 no Drummer APS para lhe dar uma resposta precisa.
Por favor, aguarde apenas alguns instantes.

Infelizmente, não é possível antecipar a entrega de 20 peças da linha 2 do pedido 17926 para o dia 08/11/2020, porque a entrega atrasaria 22 dias.

Type your message

Fonte: Autor (2020).

O Quadro 8 apresenta o resultado dos pilotos. Resumidamente, o *chatbot* respondeu corretamente a 25 das 30 perguntas recebidas, representando um percentual de acerto de 83,33%.

Quadro 8 – Resultado dos pilotos do *chatbot*

Usuário Chave	Setor	Data	Perguntas	Acertos	% Acertos	Erros	% Erros
Angélica de Carvalho dos Santos	PCP	03/11/2020	15	13	86,67 %	2	13,33 %
Marcio Silveira	Comercial	06/11/2020	15	12	80,00 %	3	20,00 %
Total			30	25	83,33 %	5	16,67 %

Fonte: Autor (2020).

Além de possibilitar a avaliação da aplicação proposta, os pilotos também serviram para identificar novas intenções e entidades que não foram compreendidas pelo *chatbot* nos testes e que poderiam ser implementadas em projetos futuros, como por exemplo:

- a) **consultar pendências:** intenção que seria utilizada principalmente pelos usuários do PCP para apurar quais são e onde estão as ordens de fabricação e compra pendentes para atender a um determinado pedido;
- b) **ordem de compra:** entidade que eliminaria a necessidade de o técnico comercial consultar o número do pedido antes de conversar com o *chatbot*, pois o cliente informa apenas o número da ordem de compra que deseja renegociar;
- c) **data de entrega:** entidade necessária para que o *chatbot* consiga responder a perguntas do tipo: “As entregas do dia X estão confirmadas?”.

5.6 PESQUISA QUALITATIVA

Concluindo o estágio de implementação e a fase II deste projeto, ao final dos pilotos, foi solicitado que os usuários chave respondessem a uma pesquisa qualitativa por meio de um formulário *on-line*, que foi criado com a ferramenta *Microsoft Forms*, conforme exemplificado na Figura 49.

Figura 49 – Seção 1 do formulário de avaliação do *chatbot*

The screenshot shows a Microsoft Forms survey titled "Avaliação do Chatbot". The subtitle reads: "Trabalho de Conclusão de Curso desenvolvido por Alexandre Pergher Amaro para obtenção do grau de Bacharel em Ciência da Computação pela Universidade de Cassia do Sul." The form is marked as "Obrigatória" (Mandatory). The questions are:

1. Qual é o seu nome completo? * (Text input field)
2. Qual é o seu setor? * (Dropdown menu)
3. Qual é o seu cargo? * (Text input field)

At the bottom, there is an "Avançar" (Next) button, a progress indicator showing "Página 1 de 7", and a footer note: "Nunca forneça sua senha. Relatar abuso".

Fonte: Autor (2020).

A finalidade dessa pesquisa era avaliar os principais aspectos do *chatbot* implementado neste projeto e validar se os objetivos iniciais deste trabalho foram alcançados.

Quando questionados com relação à satisfação geral com o *chatbot* implementado neste trabalho, ambos os usuários chave responderam à pergunta com uma nota 4, em uma escala de 1 a 5, indicando estarem satisfeitos com o resultado do projeto.

Apesar do *chatbot* não ter conseguido responder todas as perguntas dos usuários chave corretamente nos pilotos, ambos avaliaram as respostas do *chatbot* como adequadas, completas e rápidas nas questões 20, 22 e 24 do formulário, respectivamente.

Além disso, segundo as respostas dos próprios usuários chave na pesquisa qualitativa, alguns dos benefícios esperados com a utilização do *chatbot* proposto neste trabalho seriam:

- a) agilizar e padronizar as respostas para a área comercial no que diz respeito ao *follow-up* e ao aprazamento de pedidos de venda;
- b) confiar nos retornos embasados referentes às simulações de atendimento dos pedidos de venda;
- c) liberar a equipe para a execução de atividades mais importantes e estratégicas;
- d) reduzir os tempos de espera da equipe;
- e) facilitar o acesso aos dados dos pedidos de venda;
- f) disseminar a informação para diferentes níveis hierárquicos da empresa;
- g) simular diferentes cenários de antecipação de entregas com maior praticidade e assertividade.

Também vale a pena destacar que os usuários chave externaram o desejo na questão 26 do formulário de que o *chatbot* desenvolvido como um experimento neste TCC seja, de fato, implantado na empresa que eles atuam.

De acordo com a usuária chave do setor de PCP, Sra. Angélica de Carvalho dos Santos (2020), o *chatbot* desenvolvido nesse TCC “É um programa promissor que mostrou ser uma excelente proposta para automatizar tarefas repetitivas.”.

Segue também um depoimento coletado do usuário chave da área comercial, Sr. Marcio Silveira (2020), sobre o resultado deste projeto:

O *chatbot* na indústria é um desafio, porém necessário e agregador para as empresas. É uma ferramenta que agrega valor, ou seja, nos dá respostas rápidas, precisas e, principalmente, baseadas em dados que devem ser muito bem parametrizados no sistema. Nos dias de hoje, onde a informação é a grande oportunidade para o sucesso das empresas, uma ferramenta assim nos deixa mais produtivos e autônomos a ter as respostas necessárias para atender a expectativa dos clientes.

O formulário de avaliação completo com todas as respostas da pesquisa qualitativa pode ser conferido no Apêndice A.

6 CONCLUSÃO

Inicialmente, o presente trabalho tinha o objetivo de desenvolver um *chatbot* integrado aos sistemas ERP e APS de uma indústria do setor metalomecânico para responder as perguntas mais frequentes dos usuários sobre ordens de produção, visando agilizar e facilitar a obtenção de respostas pela fábrica.

Entretanto, após a execução do processo de elicitação de requisitos, em que foram entrevistados os usuários chave do projeto, mudou-se completamente o foco deste trabalho, porque pode-se observar que a grande maioria das questões que são respondidas pelo setor de PCP da empresa se referem aos pedidos de venda, principalmente relacionadas aos prazos de entrega, e que o principal cliente do setor de PCP são os técnicos da área comercial.

Além disso, os processos executados para responder essas solicitações comerciais são muito mais complexos e onerosos do que aqueles realizados para responder as dúvidas da fábrica referentes às ordens de produção. Inclusive, devido a esse fator, muitas vezes, o programador de produção acaba abdicando algumas etapas essenciais do processo e, conseqüentemente, passando um retorno desqualificado e sem embasamento para o técnico comercial. Ou seja, as respostas não são padronizadas e confiáveis.

Sendo assim, notou-se uma enorme oportunidade de melhoria no processo de comunicação interna entre esses setores da empresa, e pode-se compreender o quão importante é um processo de elicitação de requisitos bem executado para assegurar o sucesso de todo o projeto.

A propósito, a participação ativa de ambos os usuários chave neste TCC foi imprescindível para o seu sucesso, principalmente nas etapas de elicitação de requisitos, piloto e avaliação dos resultados.

Outro detalhe interessante é que a integração do *chatbot* com o sistema ERP utilizado pela empresa não foi necessária, porque os principais dados dos pedidos de venda cadastrados no ERP são importados, diariamente, pelo sistema APS que foi integrado à solução.

Também é importante citar que a metodologia aplicada neste trabalho pode ser replicada em qualquer outra organização, independentemente do segmento, adequando apenas a estratégia de integração ao SI utilizado pela empresa.

6.1 AVALIAÇÃO DOS RESULTADOS

Após a avaliação dos resultados obtidos com este trabalho, é possível afirmar que todos os objetivos explicitados na seção 1.3 foram atingidos com sucesso, e que a questão de pesquisa definida na seção 1.2 foi indiscutivelmente respondida ao longo do texto.

Essa afirmação é corroborada pelo depoimento dos usuários chave em resposta à questão 27 da pesquisa qualitativa, porque ambos disseram que a utilização do *chatbot* desenvolvido neste trabalho agilizaria e padronizaria as respostas para as perguntas mais frequentes sobre os pedidos de venda da Sazi, conforme pode ser comprovado no Apêndice A.

6.2 BENEFÍCIOS

Muito mais do que apenas responder as perguntas mais frequentes sobre os pedidos de venda da empresa que foi objeto de estudo deste TCC, a automatização desse processo por meio da utilização de um *chatbot*, tal como o que foi proposto, poderá proporcionar inúmeros benefícios, diretos e indiretos, imensuráveis para a organização analisada, inclusive além daquilo que havia sido previsto inicialmente na proposta de solução, como por exemplo:

- a) padronizar e automatizar processos repetitivos e rotineiros que são executados para consultar informações e responder as perguntas mais frequentes referentes aos pedidos de venda;
- b) agilizar o serviço de atendimento ao cliente e o retorno de dúvidas sobre os pedidos de venda e, por consequência, a operação de um modo geral;
- c) qualificar as respostas aos questionamentos sobre pedidos de venda;
- d) elevar a capacidade e a disponibilidade da organização em esclarecer dúvidas dos clientes e colaboradores internos sobre pedidos de venda;
- e) democratizar a realização de análises complexas de capacidade *versus* demanda da fábrica;
- f) fornecer suporte contínuo aos usuários com atendimento no formato 24/7 (24 horas por dia, 7 dias por semana) para sanar as dúvidas sobre pedidos de venda;

- g) atender simultaneamente as dúvidas sobre pedidos de venda de múltiplos usuários;
- h) diminuir a necessidade de consulta a alguns relatórios, planilhas e painéis de controle que podem possuir um baixo desempenho;
- i) melhorar a experiência dos usuários na utilização dos sistemas de gestão da organização;
- j) evitar as eventuais interrupções nos setores para saneamento de dúvidas sobre pedidos de venda e, conseqüentemente, melhorar o foco e aumentar a produtividade das equipes;
- k) liberar mão de obra especializada para a execução de atividades mais importantes e estratégicas;
- l) escalar a operação da empresa, ou seja, ganhar produtividade sem perder qualidade;
- m) dar mais autonomia para a área comercial negociar prazos de entrega com os clientes;
- n) aumentar o nível de confiança nas respostas obtidas para os questionamentos referente ao aprazamento de pedidos de venda, tanto por parte da equipe comercial, quanto do cliente;
- o) evitar o atravessamento de ordens de produção e outras despesas administrativas devido ao aceite de prazos de entrega impossíveis de serem realizados pela fábrica;
- p) melhorar os resultados dos indicadores de atendimento ao cliente, principalmente aqueles relativos ao cumprimento dos prazos de entrega;
- q) elevar o nível de satisfação geral dos clientes com a empresa;
- r) reduzir os custos operacionais da organização.

6.3 TRABALHOS FUTUROS

Por último, mas não menos importante, caso o Grupo Sazi demonstre interesse em implantar o *chatbot* desenvolvido nesse TCC, o próximo passo será orçar os custos desse projeto com o fornecedor de serviços do *Azure* e aprovar o investimento junto à direção da empresa. Aliás, nesse ponto vale a pena ressaltar que este trabalho pôde ser inteiramente desenvolvido com recursos gratuitos da *Microsoft*.

Se o investimento estimado for aprovado pela gestão, todos os recursos criados com uma assinatura gratuita do *Azure* terão que ser migrados para a assinatura padrão da organização, viabilizando assim a conexão do aplicativo na rede virtual privada da empresa.

No entanto, com base no excelente resultado dos pilotos, a versão *closed beta* do *chatbot* que foi gerada neste trabalho já pode ser considerada uma versão *release candidate* (RC) pronta para ser liberada em produção para os demais usuários.

Também há a oportunidade de tratar novas intenções e entidades em versões futuras do aplicativo, conforme percebido nos pilotos e indicado pelos usuários chave na pesquisa qualitativa descrita no Apêndice A, e até mesmo de integrar o *chatbot* com outros sistemas de informação que são utilizados pela organização.

Outra atividade que é de suma importância para garantir a melhoria contínua do serviço é supervisionar, periodicamente, a aprendizagem do modelo de linguagem criado no LUIS. Esse processo de aprendizado ativo pode ser realizado através da funcionalidade de revisão dos enunciados previamente analisados e selecionados pelo LUIS (MICROSOFT, 2020b), conforme evidenciado na Figura 50.

Figura 50 – Aprendizado supervisionado no LUIS

The screenshot shows the Microsoft LUIS interface for 'Modelo Luis v1.0'. The 'BUILD' tab is active, and the 'Review Endpoint Utterances' section is selected. A filter 'ConsultarAntecipacao' is applied. The interface displays two utterances with their aligned intents and options to add or delete them.

Utterance	Aligned Intent	Add/Delete
é possível antecipar a entrega de 100 peças do item 99052356 da	ConsultarAntecipacao	Consultar... ✓ ✕
linha 7 do pedido 1998 para o dia 27 / 03 / 1992 ?	ConsultarAntecipacao	Consultar... ✓ ✕

Fonte: Autor (2020).

Dessarte, o *chatbot* sugerido neste trabalho apenas para responder as perguntas mais frequentes sobre pedidos de venda poderá evoluir, de tal forma, a se tornar um assistente virtual multitarefas e ser adotado pelos demais setores da empresa.

REFERÊNCIAS

ASIMOV, Isaac. **Eu, robô**. 2. ed. Rio de Janeiro: Expressão e Cultura, 1969.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1**: Engenharia de software - qualidade de produto. Rio de Janeiro, 2003.

BEZERRA, Cicero A. **Técnicas de planejamento, programação e controle da produção e introdução à programação linear**. Curitiba: Intersaberes, 2014.

BITTENCOURT, Guilherme. **Inteligência artificial**: ferramentas e teorias. 3. ed. rev. Florianópolis: UFSC, 2006.

CAIÇARA JUNIOR, Cícero. **Sistemas integrados de gestão ERP**: uma abordagem gerencial. 2. ed. Curitiba: Intersaberes, 2015.

COLBY, Kenneth M. Modeling a paranoid mind. **Behavioral and Brain Sciences**, v. 4, n. 4, p. 515-560, 1981.

CORRÊA, Henrique L.; GIANESI, Irineu G. N.; CAON, Mauro. **Planejamento, programação e controle da produção**: MRP II / ERP. 6. ed. São Paulo: Atlas, 2019.

CRUZ, Leôncio T.; ALENCAR, Antonio J.; SCHMITZ, Eber A. **Assistentes virtuais inteligentes e chatbots**: um guia prático e teórico sobre como criar experiências e recordações encantadoras para os clientes da sua empresa. 1. ed. Rio de Janeiro: Brasport, 2018.

DA SILVA, Leandro Costa. **Gestão e melhoria de processos**: conceitos, técnicas e ferramentas. Rio de Janeiro: Brasport, 2015.

DE OLIVEIRA, Diego Bittencourt *et al.* **Desenvolvimento para dispositivos móveis**. v. 1. Porto Alegre: SAGAH, 2019.

DE SORDI, José Osvaldo; MARINHO, Bernadete de Lourdes. Integração entre sistemas: análise das abordagens praticadas pelas corporações brasileiras. **Revista Brasileira de Gestão de Negócios**, v. 9, n. 23, p. 78-93, jan./abr. 2007.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 7. ed. São Paulo: Pearson, 2018.

ERL, Thomas. **SOA**: princípios de design de serviços. São Paulo: Pearson, 2009.

HILL, Jennifer; RANDOLPH FORD, W.; FARRERAS, Ingrid G. Real conversations with artificial intelligence: a comparison between human-human online conversations and human-chatbot conversations. **Computers in Human Behavior**, v. 49, p. 245-250, ago. 2015.

IBAÑOS, Ana Maria Tramunt (org.); PAIL, Daisy Batista (org.). **Fundamentos linguísticos e computação**. Porto Alegre: EDIPUCRS, 2015.

LARMAN, Craig. **Utilizando UML e padrões**: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 11. ed. São Paulo: Pearson, 2015.

LEE, Kai-Fu. **Inteligência artificial**: como os robôs estão mudando o mundo, a forma como amamos, nos relacionamos, trabalhamos e vivemos. 1. ed. Rio de Janeiro: Globo Livros, 2019.

LINTER. **Drummer APS**. Disponível em: <http://www.linter.com.br/modulos/drummer-aps>. Acesso em: 23 nov. 2020.

LOZADA, Gisele. **Planejamento e controle da produção avançado**. Porto Alegre: SAGAH, 2017.

LUGER, George F. **Inteligência artificial**: estruturas e estratégias para a solução de problemas complexos. 4. ed. Porto Alegre: Bookman, 2004.

MARTINS, Victor Manuel Moreira. **Integração de sistemas de informação**: perspectivas, normas e abordagens. 2005. 218 f. Dissertação (Mestrado) – Universidade do Minho. Guimarães, 2005.

MAULDIN, Michael L. Chatterbots, tinymuds, and the turing test: entering the loebner prize competition. *In*: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., 1994, Seattle. **Proceedings** [...]. Seattle: AAAI, 1994. p. 16-21.

MICROSOFT. **Azure Bot Service**. Disponível em: <https://azure.microsoft.com/pt-br/services/bot-service>. Acesso em: 23 nov. 2020a.

MICROSOFT. **LUIS**. Disponível em: <https://www.luis.ai>. Acesso em: 23 nov. 2020b.

MICROSOFT. **Microsoft Bot Framework**. Disponível em: <https://dev.botframework.com>. Acesso em: 23 nov. 2020c.

MICROSOFT. **Microsoft Teams**. Disponível em: <https://www.microsoft.com/pt-br/microsoft-365/microsoft-teams>. Acesso em: 23 nov. 2020d.

MICROSOFT. **QnA Maker**. Disponível em: <http://qnamaker.ai>. Acesso em: 23 nov. 2020e.

MOELLMANN, Artur Henrique. **Aplicação da teoria das restrições no gerenciamento da cadeia de suprimentos**. 2. ed. São Paulo: Blucher, 2017.

PADILHA, Thais Cássia C.; MARINS, Fernando Augusto S. Sistemas ERP: características, custos e tendências. **Revista Produção**, v. 15, n. 1, p. 102-113, jan./abr. 2005.

- PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de software**: uma abordagem profissional. 8. ed. Porto Alegre: Bookman, 2016.
- RENZETTI, Roberto B. Carla. **Microhobby**, ano 1, n. 12, p. 24-27, 1984.
- RESHMI, S.; BALAKRISHNAN, Kannan. Implementation of an inquisitive chatbot for database supported knowledge bases. **Sādhanā**, v. 41, n. 10, p. 1173-1178, out. 2016.
- REZENDE, Denis A.; ABREU, Aline F. de. **Tecnologia da informação aplicada a sistemas de informação empresarias**: o papel estratégico da informação e dos sistemas de informação nas empresas. 9. ed. rev. e ampl. São Paulo: Atlas, 2013.
- RICH, Elaine; KNIGHT, Kevin; NAIR, Shivashankar B. **Artificial intelligence**. 3. ed. New Delhi: Tata McGraw-Hill, 2009.
- RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. 2. ed. Rio de Janeiro: Elsevier, 2004.
- SACCOL, Amarolinda Z.; PEDRON, Cristiane D.; LIBERALI NETO, Guilherme; MACADAR, Marie Anne; CAZELLA, Silvio César. Avaliação do impacto dos sistemas ERP sobre variáveis estratégicas de grandes empresas no Brasil. **RAC**, v. 8, n. 1, p. 9-34, jan./mar. 2004.
- SHAWAR, Bayan A.; ATWELL, Eric S. Using corpora in machine-learning chatbot systems. **International Journal of Corpus Linguistics**, v. 10, n. 4, p. 489-516, jan. 2005.
- SIMAS, Victor Luiz *et al.* **Desenvolvimento para dispositivos móveis**. v. 2. Porto Alegre: SAGAH, 2019.
- SIMCHI-LEVI, David; KAMINSKY, Philip; SIMCH-LEVI, Edith. **Cadeia de suprimentos**: projeto e gestão: conceitos, estratégias e estudos de caso. 3. ed. Porto Alegre: Bookman, 2010.
- SIMONETTI, Marcos Leandro. **Análise das funcionalidades de integração entre sistemas CAD/PDM e ERP**. 2002. 157 f. Dissertação (Mestrado em Engenharia Aeronáutica e Mecânica) – Instituto Tecnológico de Aeronáutica. São José dos Campos, 2002.
- SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2018.
- TANENBAUM, Andrew S.; STEEN, Maarten V. **Sistemas distribuídos**: princípios e paradigmas. 2. ed. São Paulo: Pearson, 2007.
- TUBINO, Dalvio F. **Planejamento e controle da produção**: teoria e prática. 3. ed. São Paulo: Atlas, 2017.

TURING, Alan M. Computing machinery and intelligence. **Mind**, v. 59, n. 236, p. 433-460, out. 1950.

WALLACE, Richard S. The anatomy of A.L.I.C.E. *In*: EPSTEIN, Robert; ROBERTS, Gary; BEBER, Grace (ed.). **Parsing the turing test**: philosophical and methodological issues in the quest for the thinking computer. Dordrecht: Springer, 2009. p. 181-210.

WEIZENBAUM, Joseph. ELIZA: a computer program for the study of natural language communication between man and machine. **Communications of the ACM**, v. 9, n. 1, p. 36-45, jan. 1966.

APÊNDICE A – RESPOSTAS DA PESQUISA QUALITATIVA

Quadro 9 – Respostas da pesquisa qualitativa

(continua)

Questão	Resposta 1	Resposta 2
Seção 1 – Identificação do Usuário Chave		
1. Qual é o seu nome completo?	Angélica de Carvalho dos Santos	Marcio Silveira
2. Qual é o seu setor?	PCP	Comercial
3. Qual é o seu cargo?	Líder de PCP	Técnico Comercial
Seção 2 – Pesquisa de Satisfação		
4. Qual é o seu nível de satisfação geral com o <i>chatbot</i> ?	4 – Satisfeito	4 – Satisfeito
5. Qual é o seu nível de satisfação com as perguntas que o <i>chatbot</i> é capaz de responder?	4 – Satisfeito	3 – Indiferente
6. Qual é o seu nível de satisfação com as respostas do <i>chatbot</i> ?	5 – Muito Satisfeito	3 – Indiferente
7. Qual é o seu nível de satisfação com o tempo de resposta do <i>chatbot</i> ?	5 – Muito Satisfeito	4 – Satisfeito
8. Qual é o seu nível de satisfação com a capacidade de compreensão da linguagem do <i>chatbot</i> ?	4 – Satisfeito	4 – Satisfeito
9. Qual é o seu nível de satisfação com a linguagem utilizada pelo <i>chatbot</i> ?	5 – Muito Satisfeito	3 – Indiferente

(continuação)

Questão	Resposta 1	Resposta 2
10. Qual é o seu nível de satisfação com o fluxo da conversa?	5 – Muito Satisfeito	4 – Satisfeito
11. Qual é o seu nível de satisfação em relação a sua participação nesse projeto?	5 – Muito Satisfeito	4 – Satisfeito
Seção 3 – Avaliação de Aderência		
12. Qual é a probabilidade de você utilizar esse <i>chatbot</i> no seu trabalho?	5 – Muito Provável	3 – Talvez
13. Por que você utilizaria esse <i>chatbot</i> ? Por favor, exemplifique.	Usaria o <i>chatbot</i> com minha equipe, pois padronizaria as respostas de <i>follow</i> de pedidos e informações pertinentes a aprazamento de pedidos. Além disso, a facilidade do uso permite a disseminação de informação para diferentes níveis hierárquicos, dando espaço na rotina para execução de estratégias prioritárias.	Para muitas consultas, alguns formatos em relatórios são mais fáceis de analisar ou ter o entendimento. Mas o <i>chatbot</i> é muito mais prático e assertivo para os ambientes de simulação de cenários (antecipações de entregas, entregas parciais antecipadas, etc.). Sendo assim, para simulações de cenários é uma ferramenta bem útil.

(continuação)

Questão	Resposta 1	Resposta 2
14. Na sua opinião, quais seriam os benefícios da utilização desse <i>chatbot</i> ?	Padronização, redução nos tempos de espera e facilidade de uso.	Entendo que a ferramenta tem vários benefícios e na medida que amadureça traga ainda mais, mas em minha opinião, o grande ganho é ter respostas embasadas no sistema. Com o crescimento da empresa, pessoas, etc., fica cada vez mais difícil ter uma resposta assertiva das entregas. O time tem opiniões e <i>feelings</i> diferentes, e isso não traz um padrão nas consultas. A ferramenta tem um padrão, e muito além disso, este padrão é embasado no sistema (capacidade, demanda, IROG, disponibilidade de matéria prima, etc.).
Seção 4 – Avaliação das Perguntas		
15. O <i>chatbot</i> lhe respondeu todas as perguntas corretamente?	Não.	Não.
16. Quais perguntas o <i>chatbot</i> não soube lhe responder?	Quais pedidos iriam atrasar em um dia específico.	Para consultas de itens/pedidos que já foram atendidos ele retornou que não localizou, mas poderia retornar que já foi

(continuação)

Questão	Resposta 1	Resposta 2
		atendido, trazendo data de atendimento, nota fiscal, etc. Não conseguiu distinguir a quebra em dias ao ser solicitado se as entregas do cliente X estavam confirmadas para o dia Y.
17. Essas perguntas são relevantes para o seu trabalho? Por quê?	Sim, para fazer <i>follow-up</i> com os clientes, visto que é um requisito da norma ISO 9001.	A pergunta sobre entregas do dia é muito importante, pois através dela sabemos os impactos que teremos nos embarques dos clientes.
18. Por favor, marque todas as intenções que você testou no piloto. Intenção é o desejo do usuário com a mensagem.	a) Cumprimentar (Olá!); b) Consultar Antecipação de Pedido (É possível antecipar a entrega do pedido 12345 para o dia 21/05/2022?); c) Consultar Atraso de Pedido (O pedido 12345 será entregue no prazo?); d) Consultar Estoque de Pedido (O pedido 12345 já está pronto no estoque?); e) Consultar Informação de Pedido (Qual é data de entrega reprogramada do pedido 12345?);	a) Cumprimentar (Olá!); b) Consultar Antecipação de Pedido (É possível antecipar a entrega do pedido 12345 para o dia 21/05/2022?); c) Consultar Atraso de Pedido (O pedido 12345 será entregue no prazo?); d) Consultar Estoque de Pedido (O pedido 12345 já está pronto no estoque?); e) Consultar Informação de Pedido (Qual é data de entrega reprogramada do pedido 12345?);

(continuação)

Questão	Resposta 1	Resposta 2
	f) Despedir (Tchau!).	f) Despedir (Tchau!).
19. Por favor, marque todas as entidades que você testou no piloto. Entidade é qualquer dado da mensagem que complementa a intenção do usuário.	a) Nome do Cliente; b) Código do Cliente; c) Código do Item; d) Número do Pedido; g) Data.	a) Nome do Cliente; b) Código do Cliente; c) Código do Item; d) Número do Pedido; e) Linha do Pedido; f) Quantidade; g) Data.
Seção 5 – Avaliação das Respostas		
20. As respostas do <i>chatbot</i> são adequadas?	Sim.	Sim.
21. Por que você não considera as respostas do <i>chatbot</i> adequadas?	Não se aplica.	Não se aplica.
22. As respostas do <i>chatbot</i> são completas?	Sim.	Sim.
23. Por que você não considera as respostas do <i>chatbot</i> completas?	Não se aplica.	Não se aplica.
24. As respostas do <i>chatbot</i> são rápidas?	Sim.	Sim.
25. Por que você não considera as respostas do <i>chatbot</i> rápidas?	Não se aplica.	Não se aplica.
Seção 6 – Trabalhos Futuros		
26. Você gostaria que esse <i>chatbot</i> fosse implantado na sua empresa?	Sim.	Sim.

(continuação)

Questão	Resposta 1	Resposta 2
27. Por que você gostaria que esse <i>chatbot</i> fosse implantado na sua empresa?	Para agilizar os retornos para o comercial, de forma mais padronizada, liberando mais tempo da equipe para atividades mais importantes.	<p>Principalmente para termos uma maneira rápida, padrão e confiável para simularmos situações de atendimento.</p> <p>Exemplos:</p> <p>a) Posso vender para X data um item com Y peças?</p> <p>b) Posso antecipar a entrega do pedido/item X?</p> <p>c) Posso acrescentar X peças no pedido Y já em andamento e atender no mesmo prazo?</p>
28. Por que você não gostaria que esse <i>chatbot</i> fosse implantado na sua empresa?	Não se aplica.	Não se aplica.
29. Você tem alguma sugestão de melhoria para o <i>chatbot</i> ?	Identificar as faltas dos pedidos (relatório de pendências por pedido).	Segregar o dia para ter respostas da carteira diária e ter a opção de trabalhar na linguagem do cliente (ordem de compra).

(continuação)

Questão	Resposta 1	Resposta 2
Seção 7 – Conclusão		
<p>30. Qual é a sua opinião sobre o <i>chatbot</i> desenvolvido nesse TCC? Por favor, deixe aqui o seu depoimento sobre o resultado desse projeto.</p>	<p>É um programa promissor que mostrou ser uma excelente proposta para automatizar tarefas repetitivas.</p>	<p>O <i>chatbot</i> na indústria é um desafio, porém necessário e agregador para as empresas. É uma ferramenta que agrega valor, ou seja, nos dá respostas rápidas, precisas e, principalmente, baseadas em dados que devem ser muito bem parametrizados no sistema. Nos dias de hoje, onde a informação é a grande oportunidade para o sucesso das empresas, uma ferramenta assim nos deixa mais produtivos e autônomos a ter as respostas necessárias para atender a expectativa dos clientes.</p>
<p>31. Gostaria de registrar mais alguma consideração final?</p>	<p>Não.</p>	<p>Só quero agradecer a oportunidade em participar desta fase do TCC, onde foi possível apresentar o trabalho do Comercial, do qual é um ponto de ligação muito forte entre o cliente e a</p>

(conclusão)

Questão	Resposta 1	Resposta 2
		<p>fábrica, tendo que interagir, ter respostas e suporte para ambos os lados. Além disso, foi gratificante conhecer uma pequena parte da automação que se faz necessária para uma simples resposta, imaginando o desafio para atender tantos questionamentos que temos todos os dias. É muito bom poder conhecer e participar das melhorias das quais vivenciamos, onde só assim não ficamos parados no tempo. Obrigado pelo convite.</p>